



ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



Bài 2

Giới thiệu về mạng tích chập

Conv Neural Networks

Mục lục

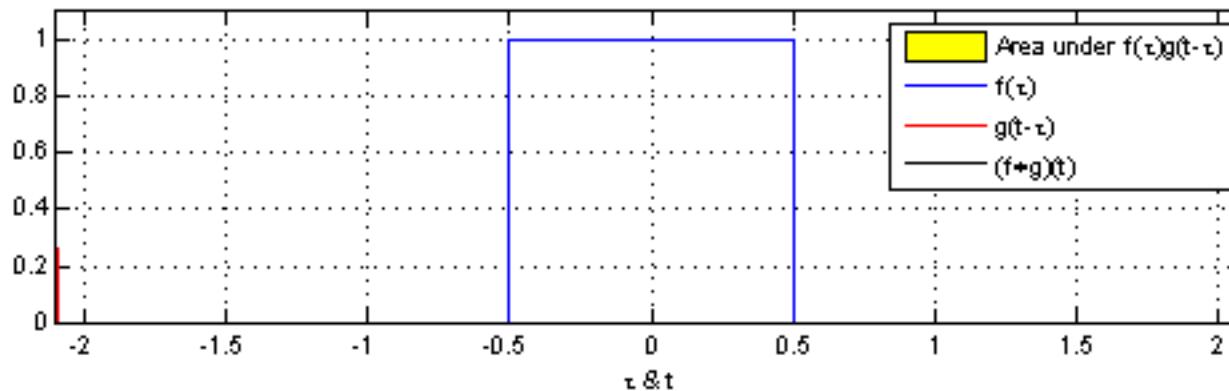
- Giới thiệu tổng quan
- Lịch sử CNN
- Các lớp trong mạng CNN
- Một vài mạng CNN cơ bản

Giới thiệu tổng quan

Hàm tích chập (Convolution)

- Trong toán học, tích chập là 1 phép toán tích phân đặc biệt thực hiện đối với 2 hàm số f và g , kết quả cho ra 1 hàm số thứ 3 thể hiện khuôn dạng của một hàm được chỉnh sửa bởi hàm còn lại.

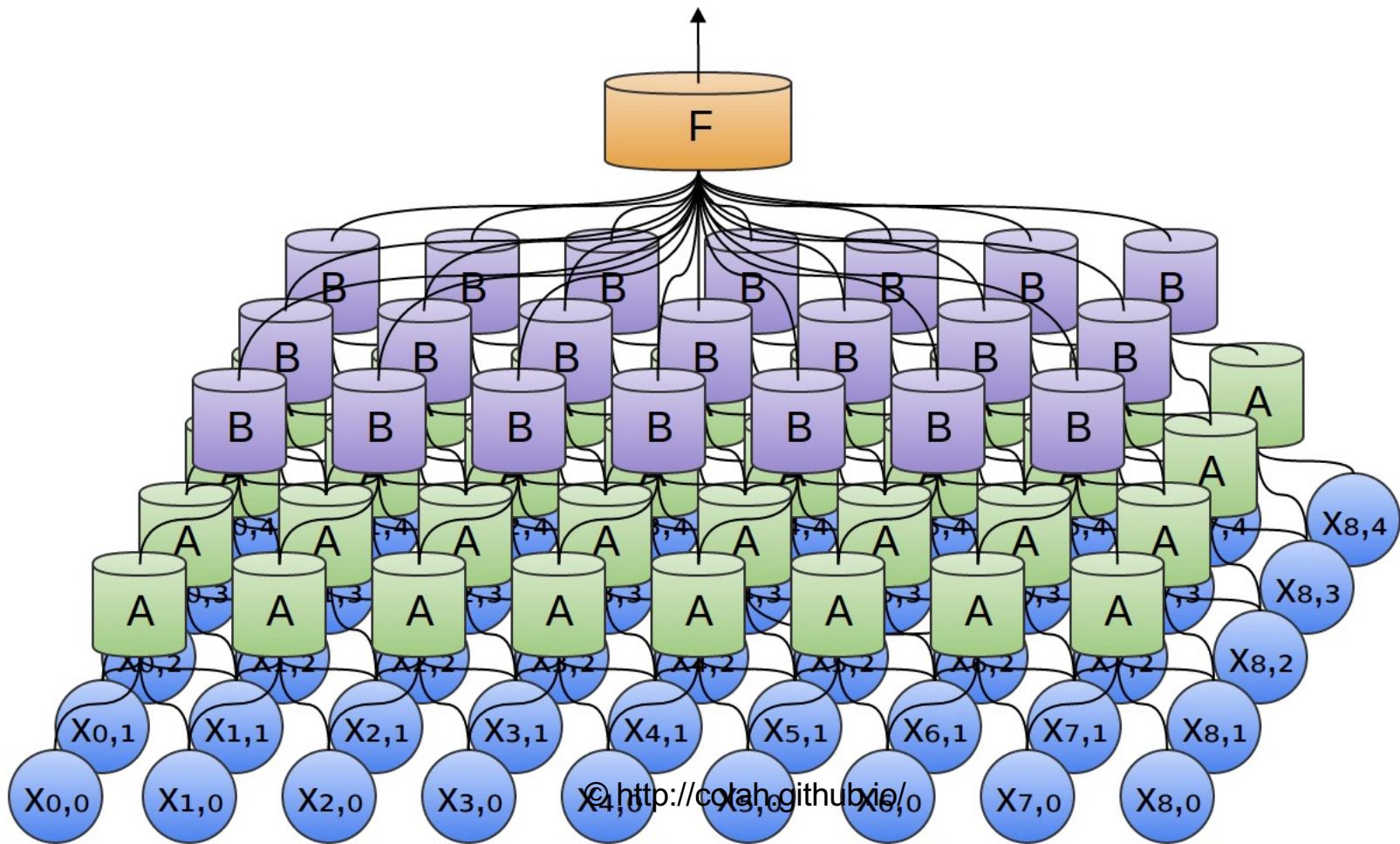
$$\begin{aligned}(f * g)(t) &\stackrel{\text{def}}{=} \int_{-\infty}^{\infty} f(\tau) g(t - \tau) d\tau \\&= \int_{-\infty}^{\infty} f(t - \tau) g(\tau) d\tau.\end{aligned}$$



Mạng nơ ron tích chập (CNN)

- Xếp chồng nhiều tầng tích chập
- Khai thác đặc trưng cấu trúc “spatial” của dữ liệu
- Sử dụng nhiều bản sao giống hệt nhau của cùng một khối nơ ron
 - Số lượng các nơ ron trong mạng lớn
 - Số lượng các tầng, độ sâu của mạng lớn
 - Nhưng số lượng các trọng số cần phải học của mô hình nhỏ hơn đáng kể

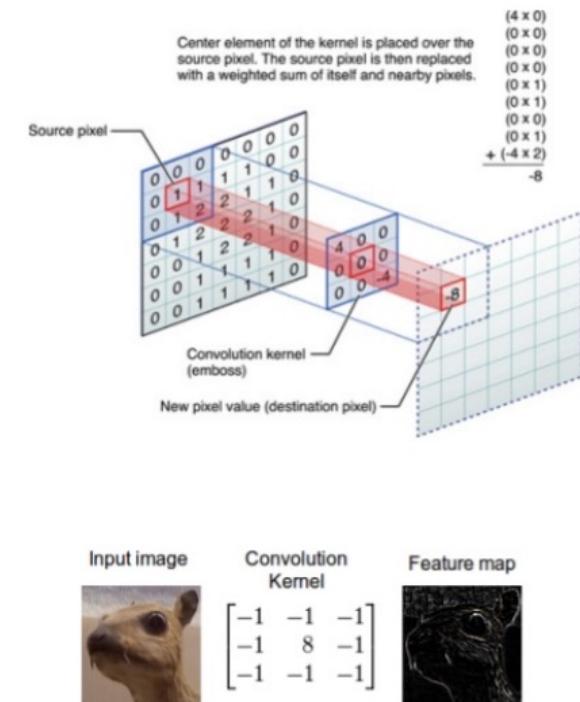
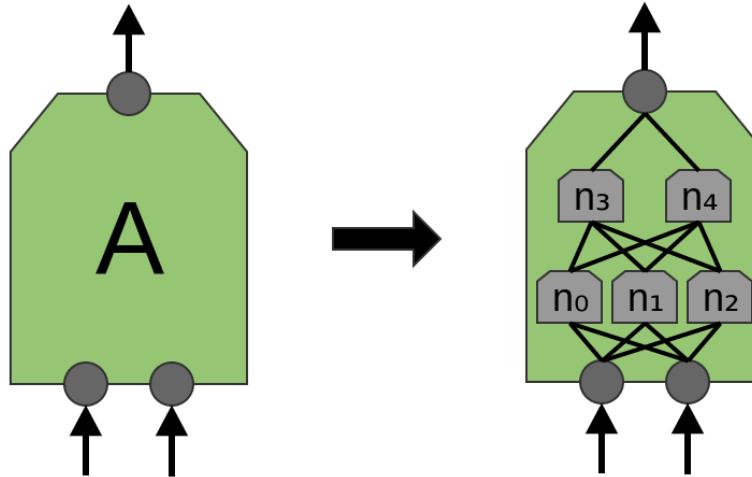
Hình dung về CNN trong không gian 2 chiều



© http://colah.github.io/

Mỗi khối nơ ron là một mạng nơ ron nhỏ

- Blocks are called filters or kernels



Tích chập trong xử lý ảnh



input

Kernel for blurring

0.062 5	0.125	0.062 5
0.125	0.25	0.125
0.062 5	0.125	0.062 5



`tf.nn.conv2d`



output

© <http://web.stanford.edu/class/cs20si>

Cách thức hoạt động của filter

1 <small>$\times 1$</small>	1 <small>$\times 0$</small>	1 <small>$\times 1$</small>	0	0
0 <small>$\times 0$</small>	1 <small>$\times 1$</small>	1 <small>$\times 0$</small>	1	0
0 <small>$\times 1$</small>	0 <small>$\times 0$</small>	1 <small>$\times 1$</small>	1	1
0	0	1	1	0
0	1	1	0	0

Image

4		

Convolved Feature

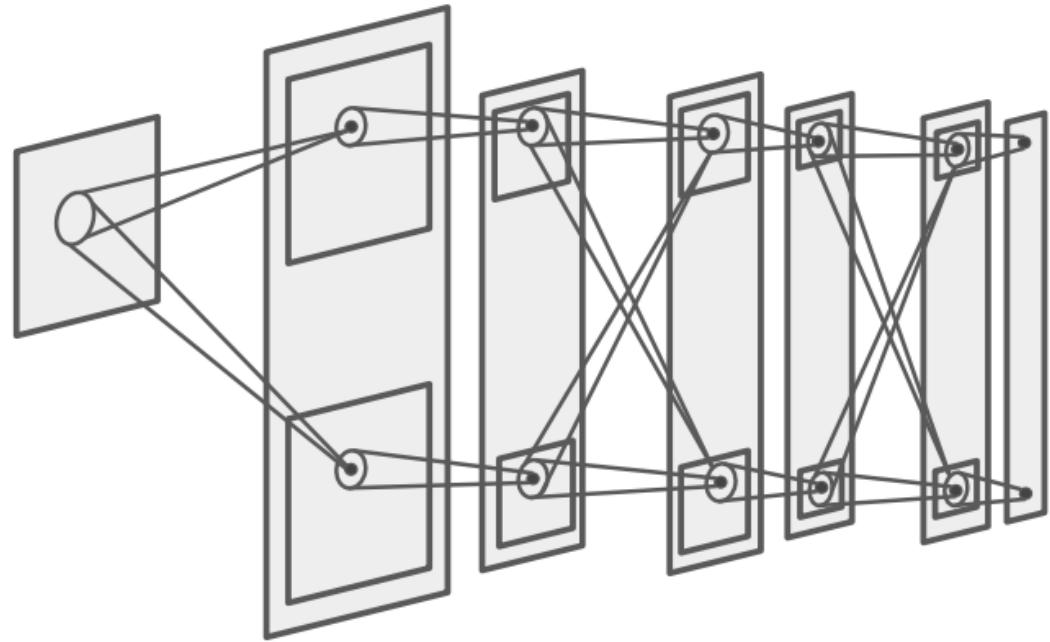
© <http://deeplearning.stanford.edu/>

Lịch sử CNN

Lịch sử CNNs

Neocognitron [Fukushima 1980]

“sandwich” architecture (SCSCSC...)
simple cells: modifiable parameters
complex cells: perform pooling

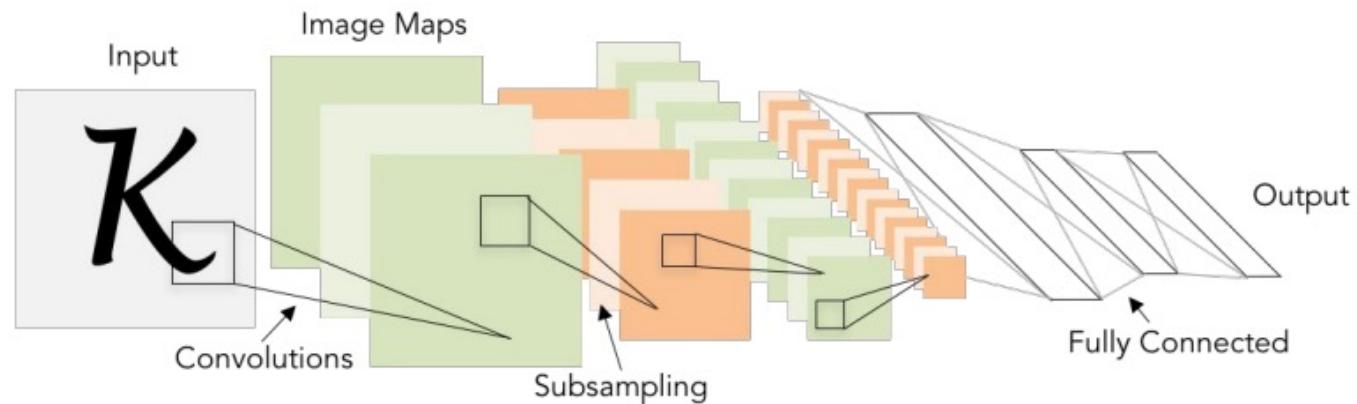


- Ý tưởng CNNs xuất phát đầu tiên từ công trình của Fukushima năm 1980

Lịch sử CNNs

Gradient-based learning applied to document recognition

[LeCun, Bottou, Bengio, Haffner 1998]



- Năm 1998, LeCun áp dụng BackProp huấn luyện mạng CNNs cho bài toán nhận dạng văn bản

Lịch sử CNNs

ImageNet Classification with Deep Convolutional Neural Networks
[Krizhevsky, Sutskever, Hinton, 2012]

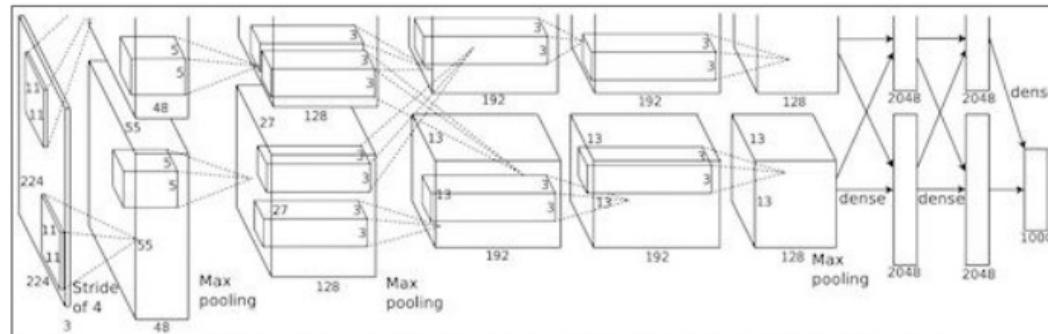


Figure copyright Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, 2012. Reproduced with permission.

“AlexNet”

- Năm 2012, CNNs gây tiếng vang lớn khi vô địch cuộc thi ILSRC 2012, vượt xa phương pháp đứng thứ 2 theo cách tiếp cận thị giác máy tính truyền thống.

Lịch sử CNNs

Classification



Retrieval

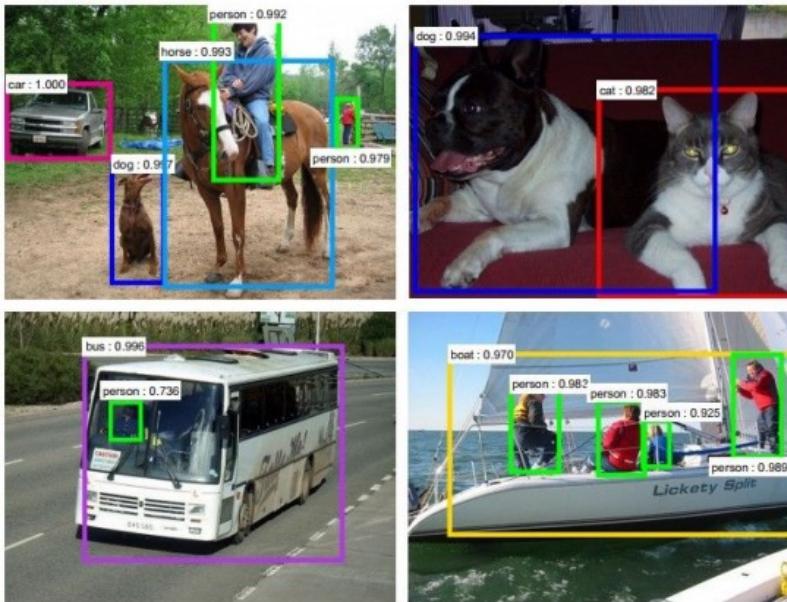


Figures copyright Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, 2012. Reproduced with permission.

- Hiện nay CNNs ứng dụng khắp nơi, ví dụ trong bài toán phân loại ảnh, truy vấn ảnh

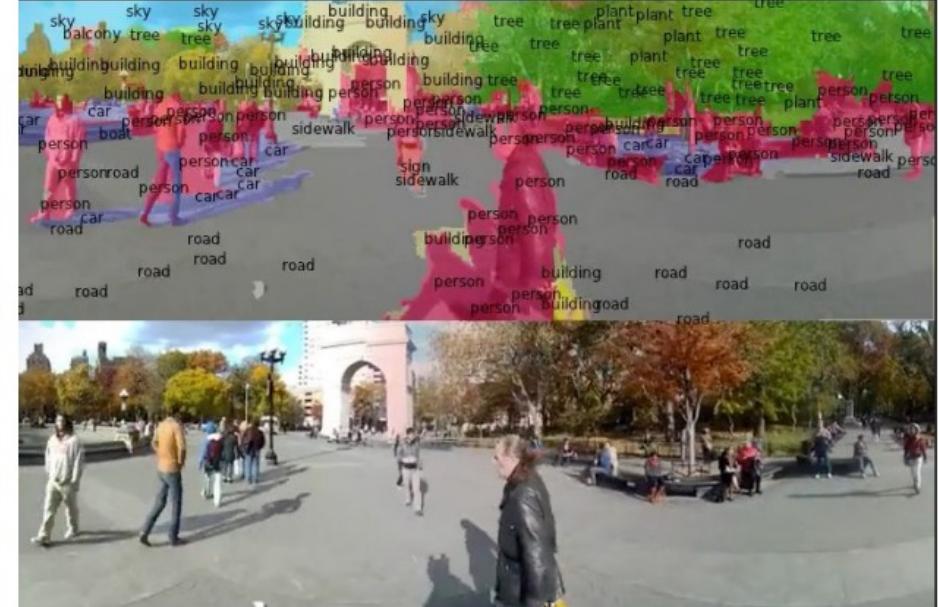
Lịch sử CNNs

Detection



[Faster R-CNN: Ren, He, Girshick, Sun 2015]

Segmentation



[Farabet et al., 2012]

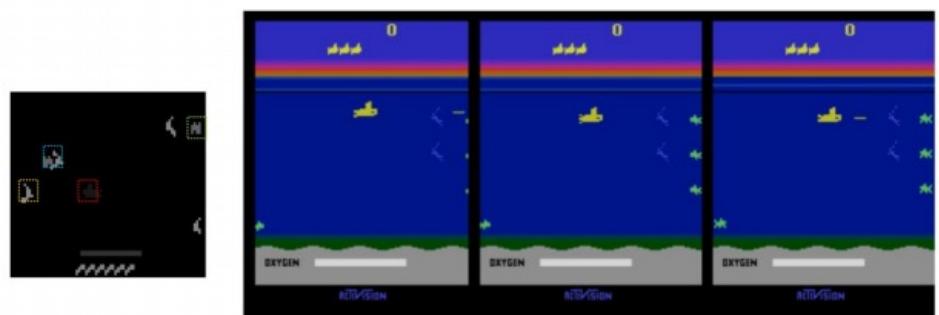
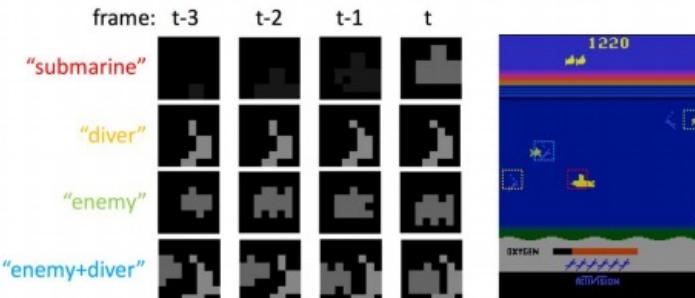
- Ứng dụng CNNs trong bài toán phát hiện đối tượng, phân đoạn ảnh

Lịch sử CNNs



Images are examples of pose estimation, not actually from Toshev & Szegedy 2014. Copyright Lane McIntosh.

[Toshev, Szegedy 2014]



[Guo et al. 2014]

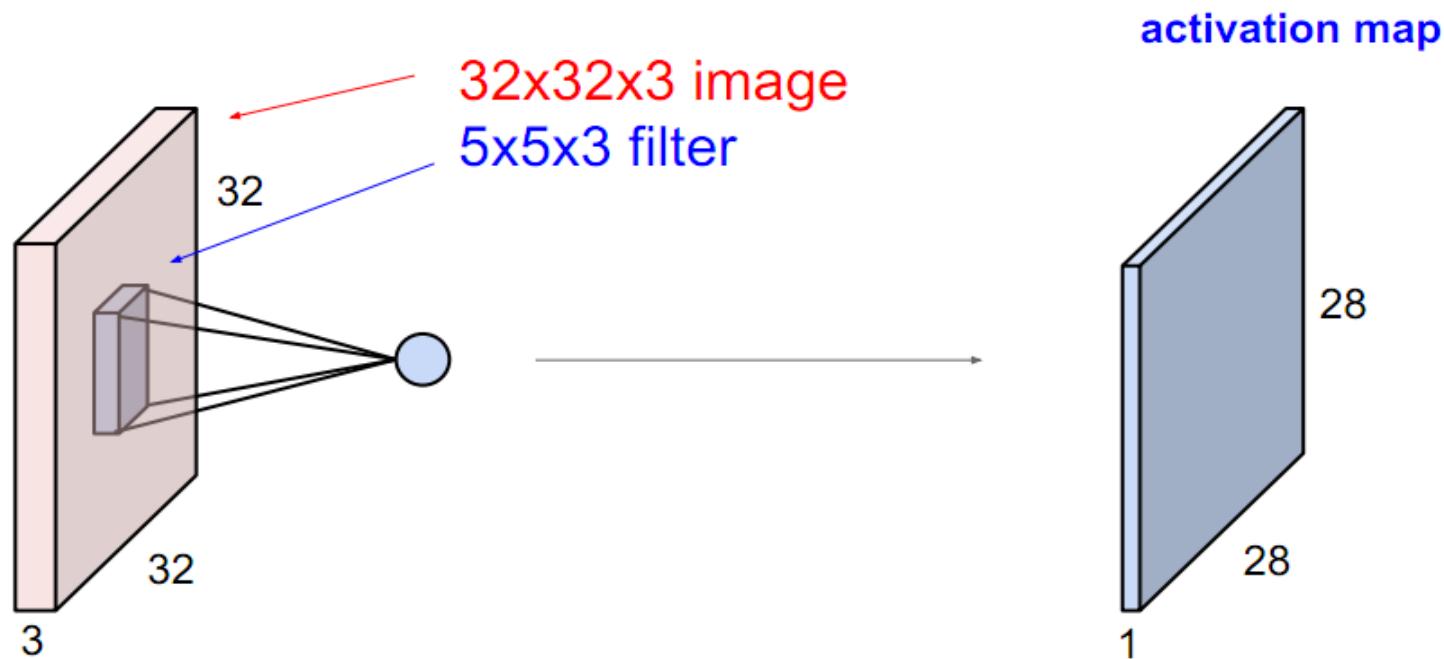
Figures copyright Xiaoxiao Guo, Satinder Singh, Honglak Lee, Richard Lewis, and Xiaoshi Wang, 2014. Reproduced with permission.

- Ứng dụng CNNs trong nhận dạng dáng người (human pose), trong trò chơi...

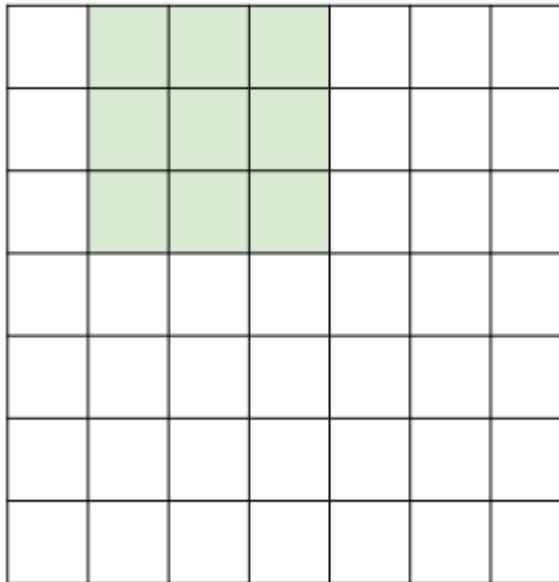
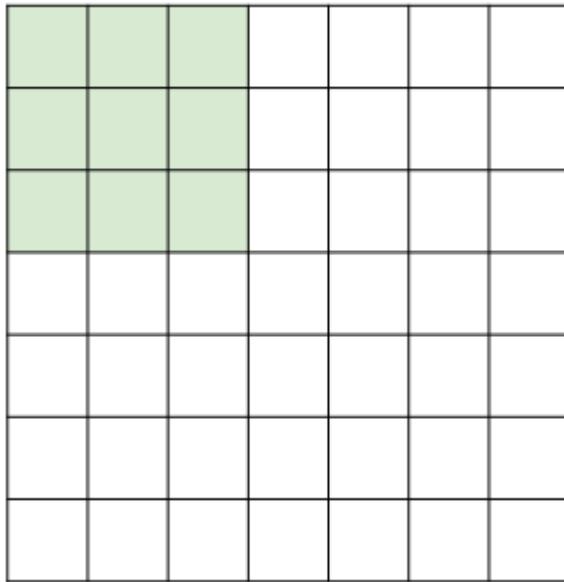
Các lớp trong mạng CNN

Lớp tích chập

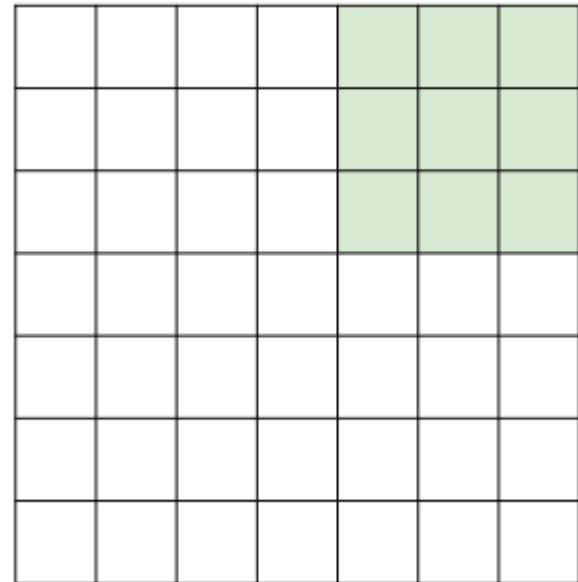
- Khác với nơ-ron kết nối đầy đủ, mỗi nơ-ron tích chập (filter) chỉ kết nối cục bộ với dữ liệu đầu vào
- Nơ-ron tích chập trượt từ trái sang phải và từ trên xuống dưới khôi dữ liệu đầu vào và tính toán để sinh ra một bản đồ kích hoạt (activation map)
- Chiều sâu của nơ-ron tích chập bằng chiều sâu của khối dữ liệu đầu vào



Lớp tích chập

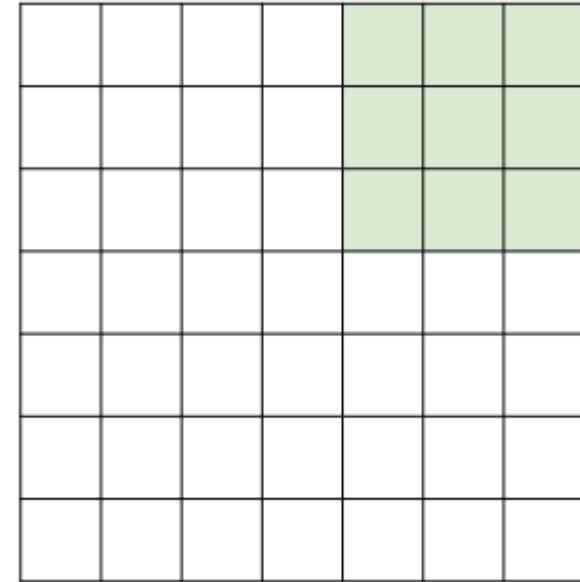
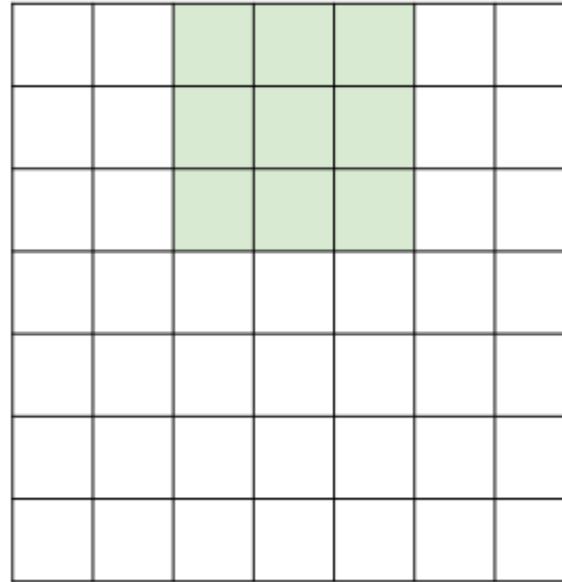
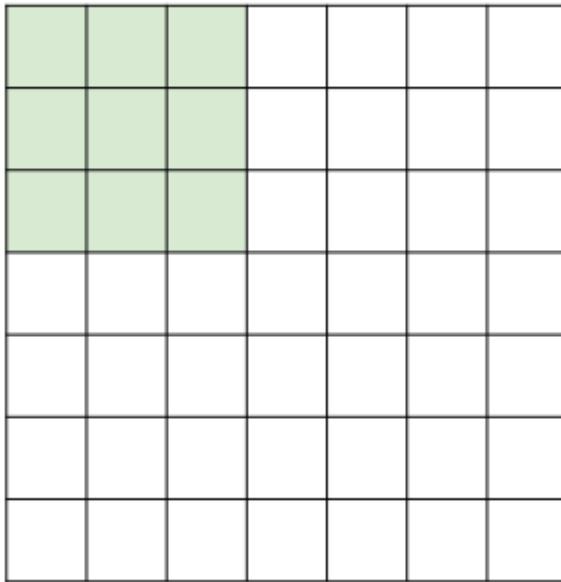


...



- Bước nhảy stride = 1
- Đầu vào kích thước 7×7 , nơ-ron kích thước 3×3
- Đầu ra kích thước 5×5

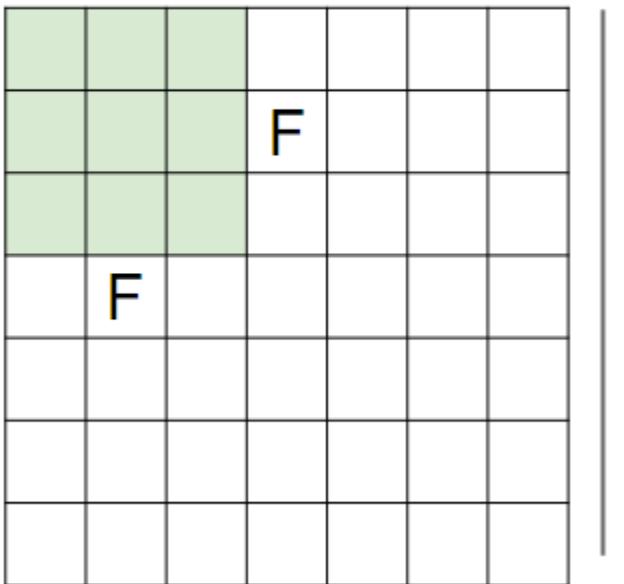
Lớp tích chập



- Bước nhảy stride = 2
- Đầu vào kích thước 7×7 , nơ-ron kích thước 3×3
- Đầu ra kích thước 3×3

Lớp tích chập

N



N

Output size:
 $(N - F) / \text{stride} + 1$

e.g. $N = 7$, $F = 3$:

$$\text{stride } 1 \Rightarrow (7 - 3)/1 + 1 = 5$$

$$\text{stride } 2 \Rightarrow (7 - 3)/2 + 1 = 3$$

$$\text{stride } 3 \Rightarrow (7 - 3)/3 + 1 = 2.33 : \backslash$$

Lớp tích chập

- Để bảo toàn kích thước thường thêm viền bởi các số 0 (zero padding).
- Ví dụ: đầu vào kích thước 7×7 , nơ-ron kích thước 3×3 , bước nhảy stride 1, padding viền độ rộng 1.
- Khi đó kích thước đầu ra là 7×7

0	0	0	0	0	0	0			
0									
0									
0									
0									

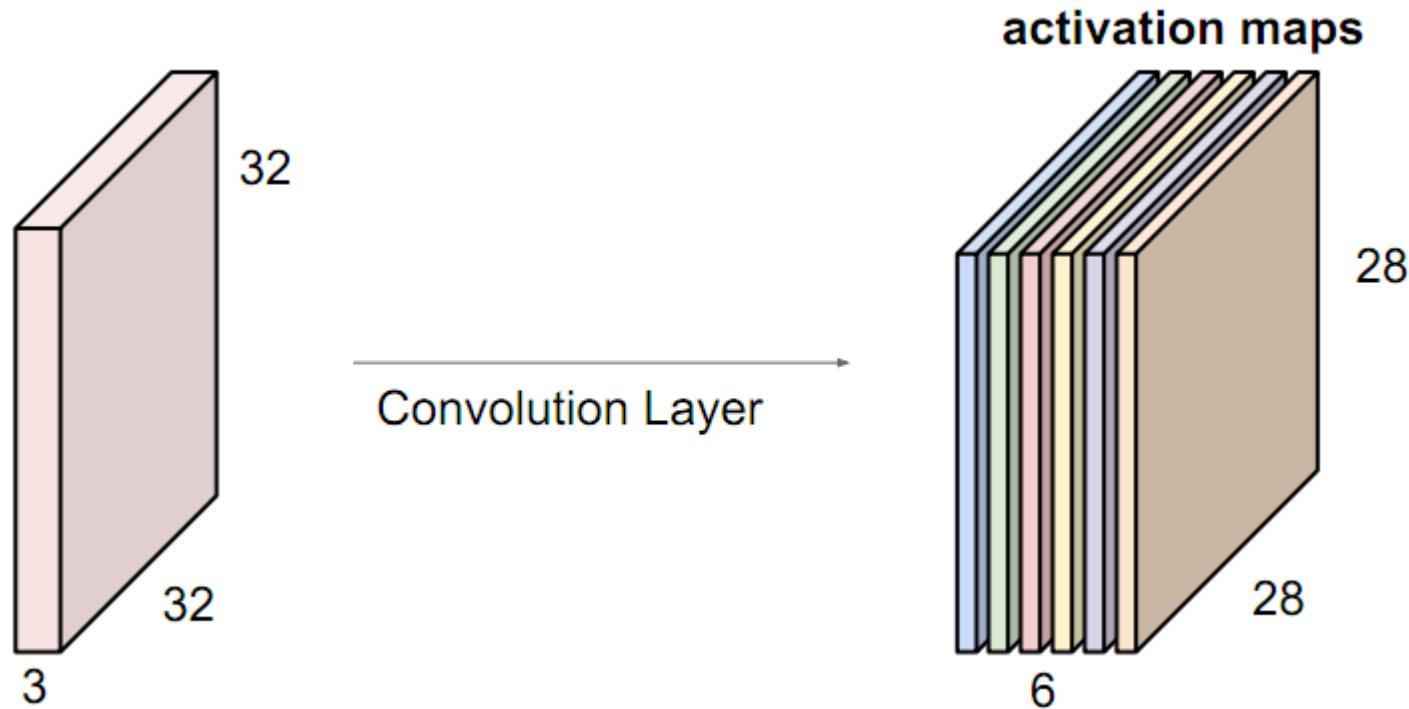
Lớp tích chập

- Giả sử có thêm nơ-ron tích chập khác thì nó cũng hoạt động tương tự và sinh ra bản đồ kích hoạt thứ hai
- Lưu ý trọng số của các nơ-ron tích chập là khác nhau



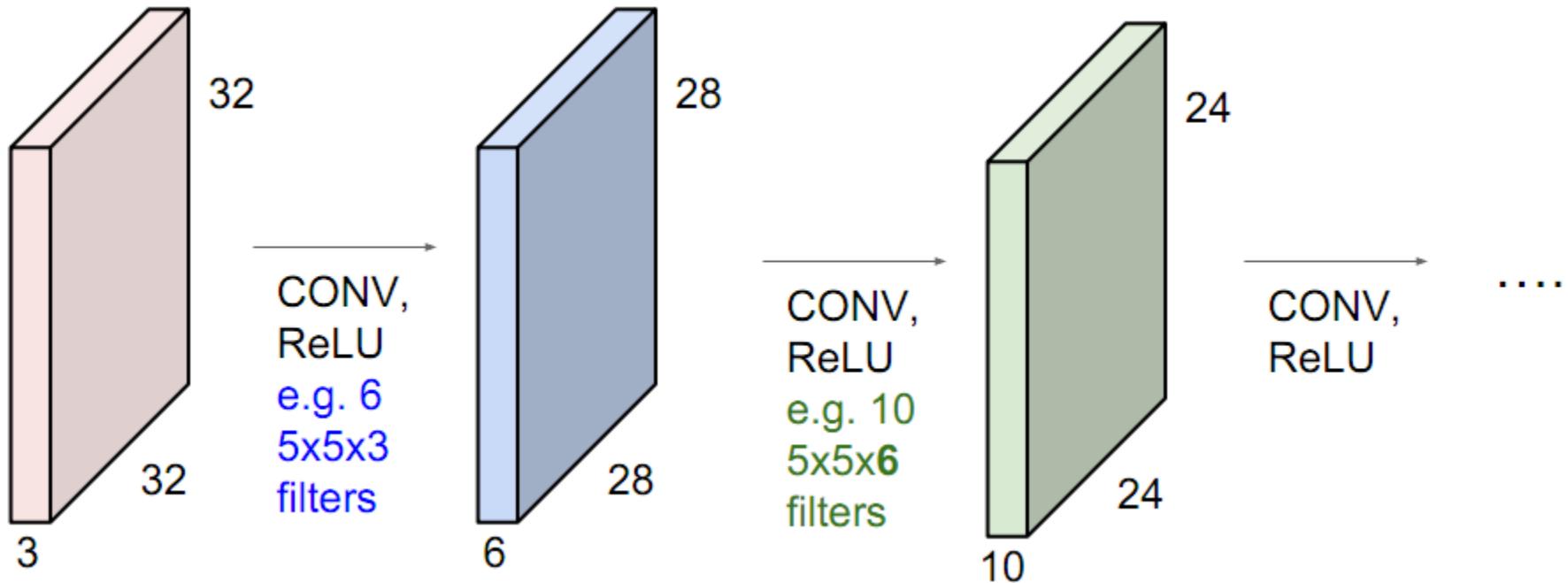
Lớp tích chập

- Giả sử có 6 nơ-ron tích chập sẽ sinh ra 6 bản đồ kích hoạt
- Các bản đồ kích hoạt ghép với nhau thành một “ảnh mới”



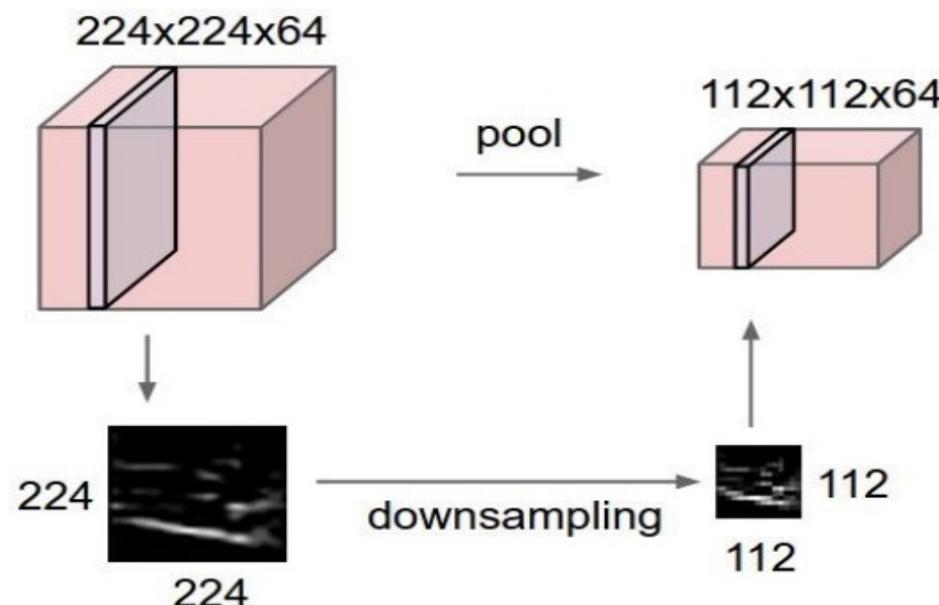
CNNs

- Mạng nơ-ron tích chập là một dãy các lớp tích chập nối liên tiếp nhau xen kẽ bởi các hàm kích hoạt (ví dụ ReLU)

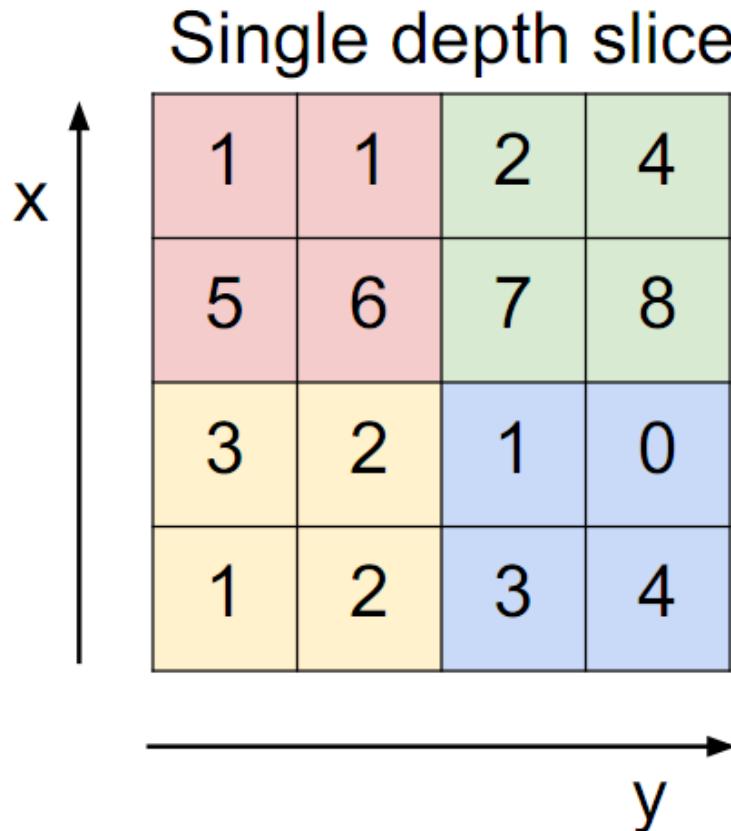


Lớp gộp (pooling layer)

- Giúp giảm độ phân giải của khối dữ liệu để giảm bộ nhớ và khôi lượng tính toán
- Hoạt động độc lập trên từng bản đồ kích hoạt
- Lớp gộp max pooling giúp mạng biểu diễn bất biến đối với các thay đổi tịnh tiến (translation invariance) hoặc biến dạng (deformation invariance) của dữ liệu đầu vào



Lớp gộp max pooling

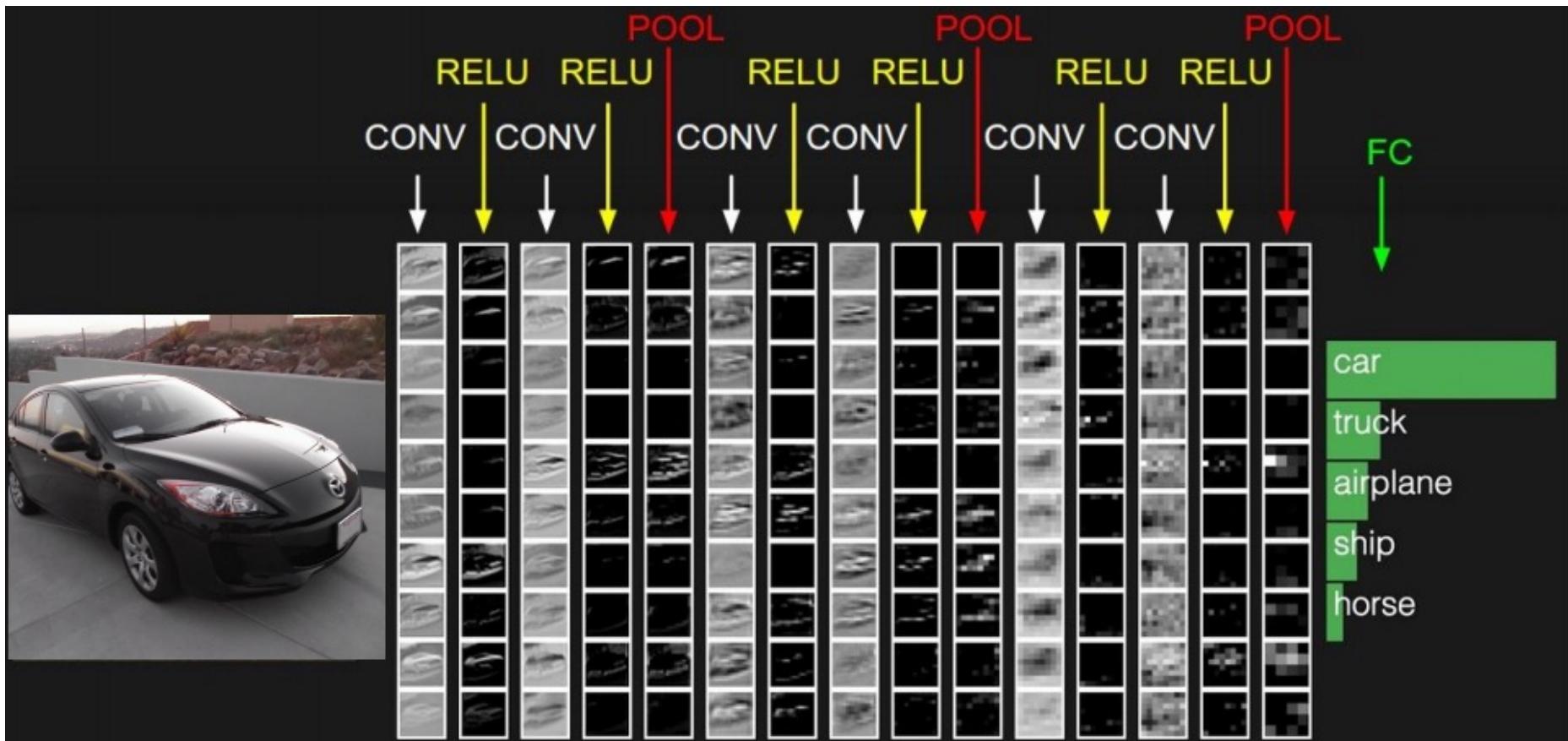


max pool with 2x2 filters
and stride 2

The resulting 2x2 output matrix has values 6 and 8 in the top-left and bottom-right positions respectively, while the other two positions are 0.

6	8
3	4

CNNs

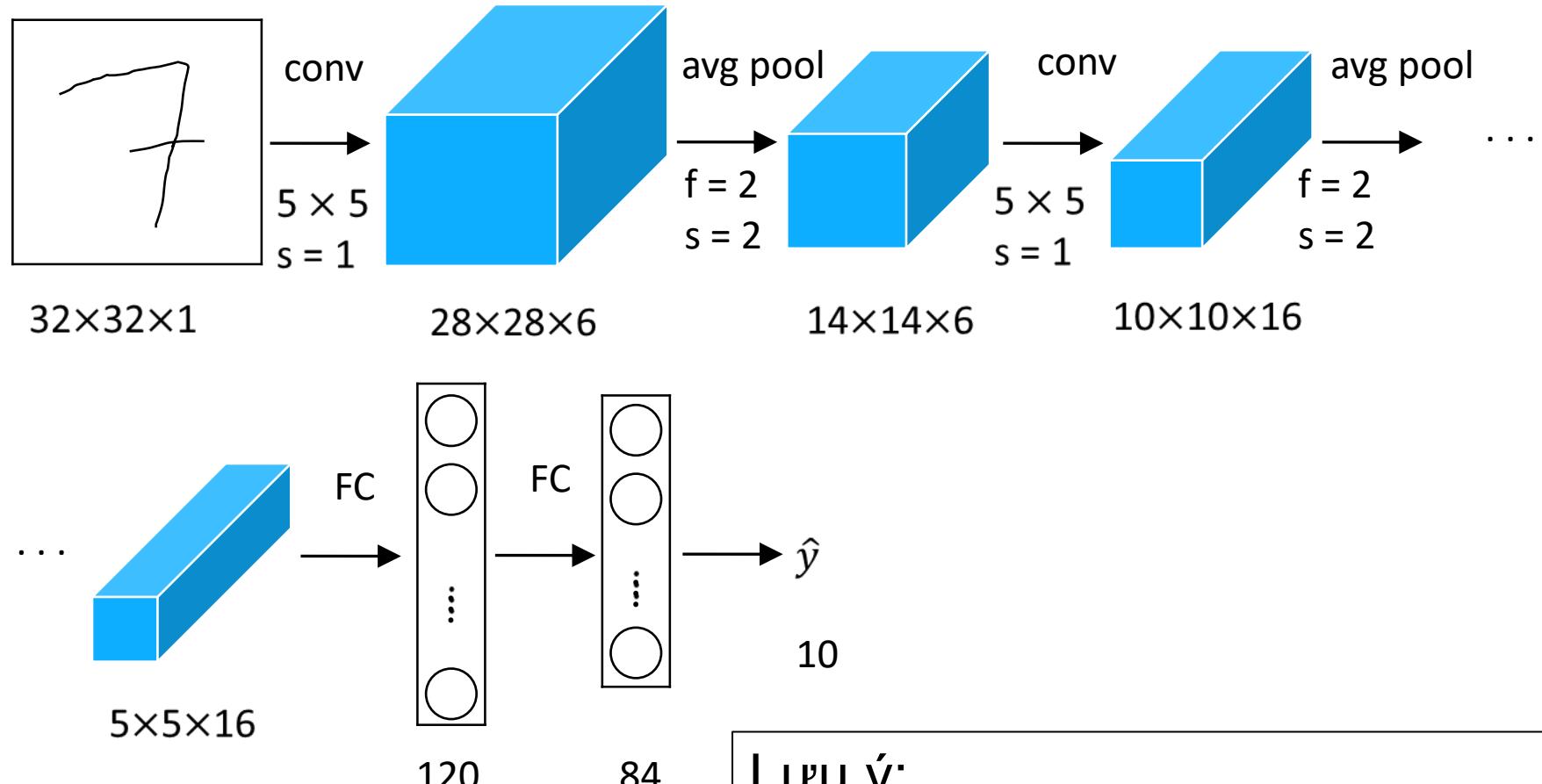


Một vài mạng CNN cơ bản

Một số mạng CNNs cơ bản

- LeNet-5
- AlexNet
- VGG
- GoogleNet
- ResNet

LeNet-5



Lưu ý:
Output size = $(N+2P-F)/\text{stride} + 1$

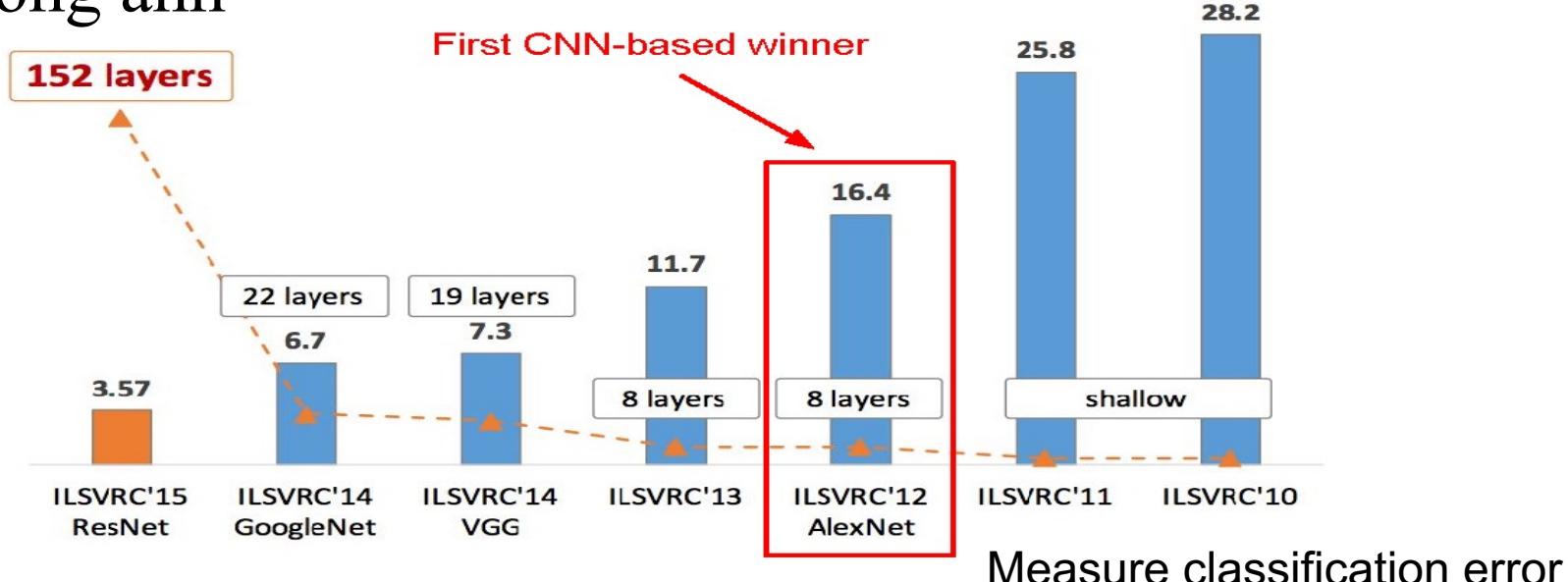
AlexNet

- ImageNet Classification with Deep Convolutional Neural Networks - Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton; 2012
- Một trong những mạng CNNs lớn nhất tại thời điểm đó
- Có 60M tham số so với 60k tham số LeNet-5

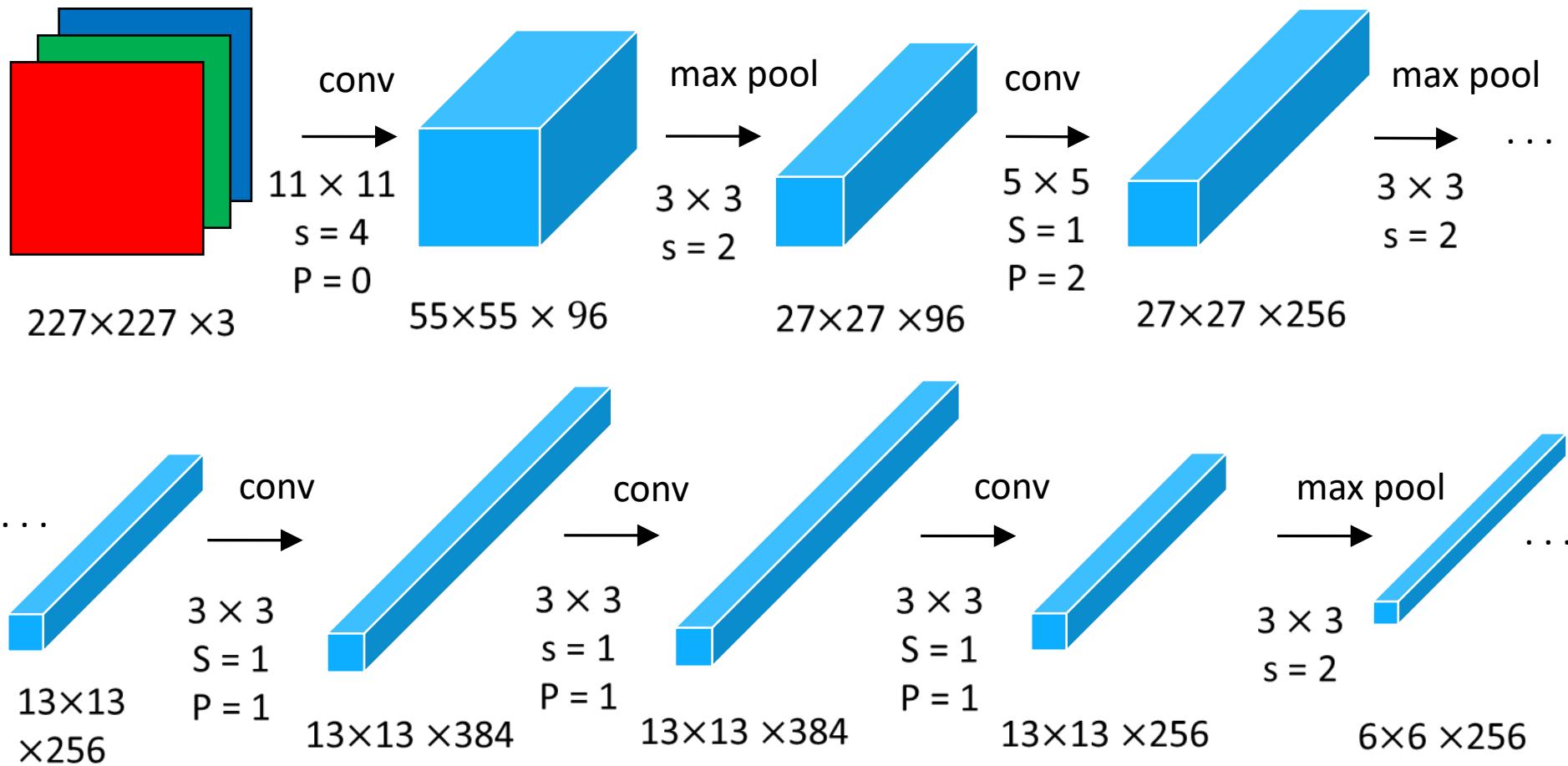
[Krizhevsky et al., 2012]

ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners

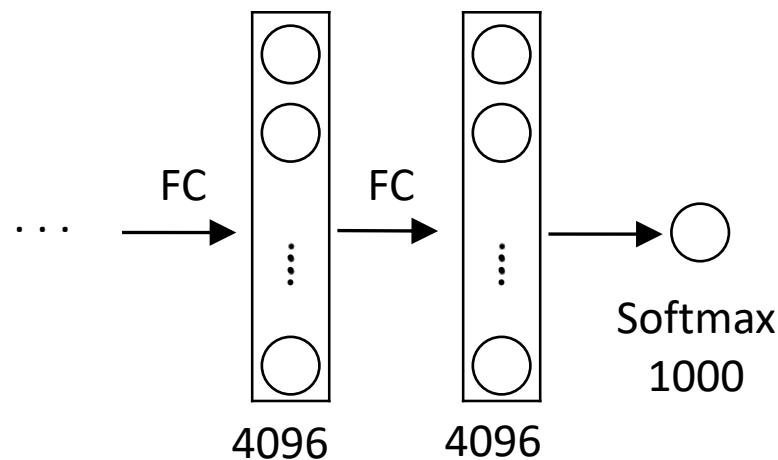
- “Olympics” thường niên về lĩnh vực thị giác máy tính.
- Các teams khắp thế giới thi đấu với nhau để xem ai là người có mô hình CV tốt nhất cho các bài toán như phân loại ảnh, định vị và phát hiện đối tượng trong ảnh



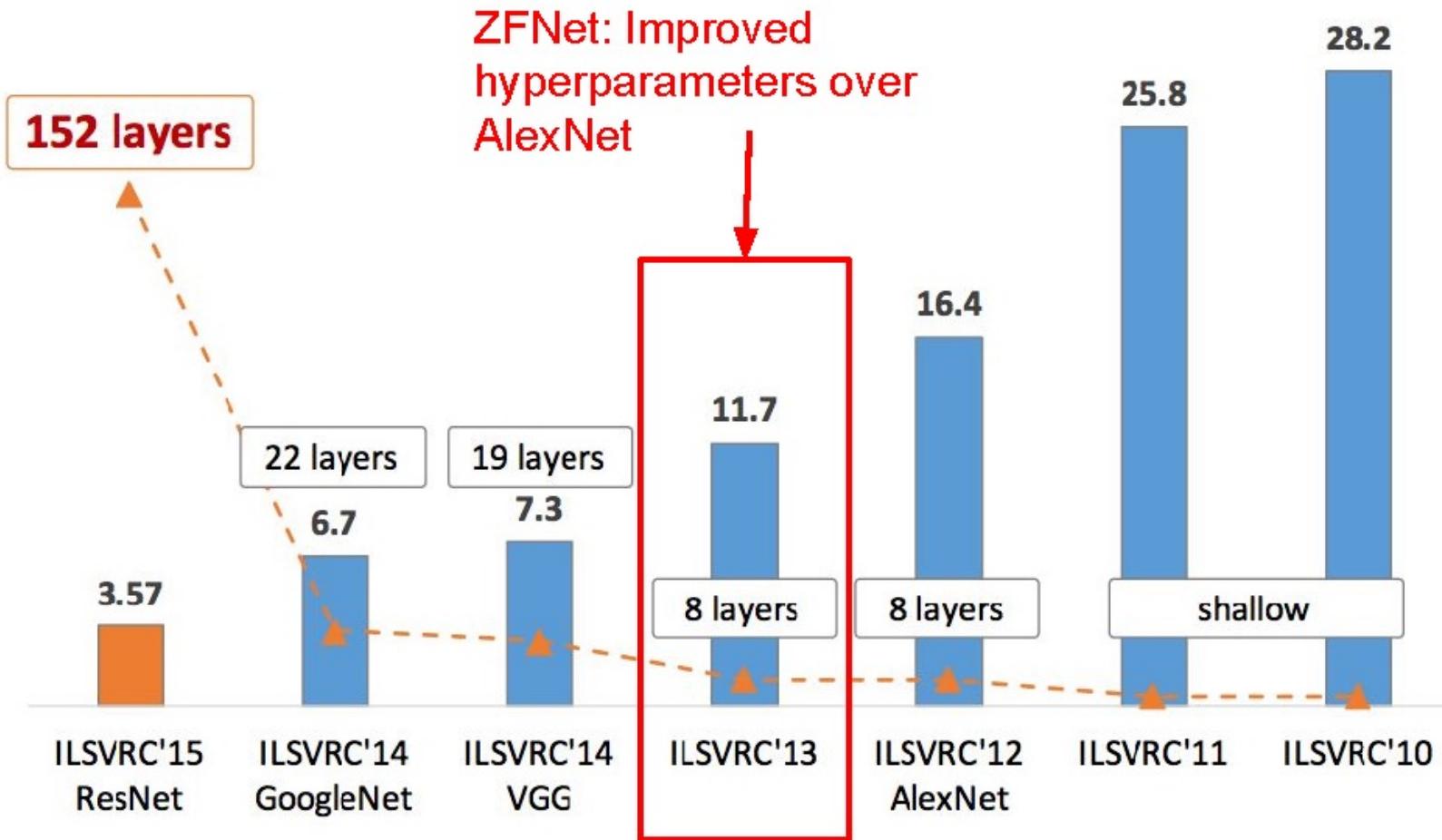
AlexNet



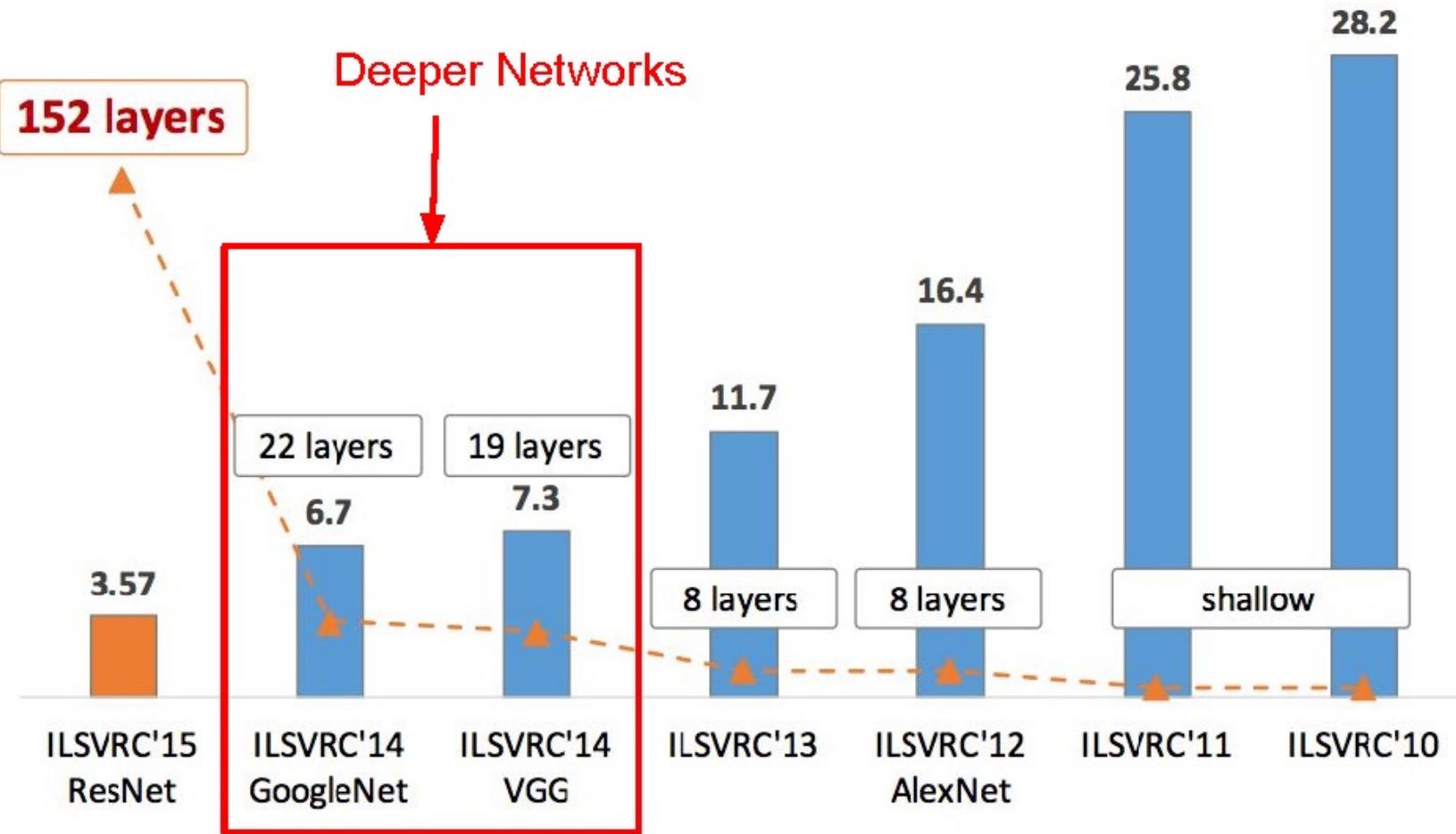
AlexNet



ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners



ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners



VGGNet

- Very Deep Convolutional Networks For Large Scale Image Recognition - Karen Simonyan and Andrew Zisserman; 2015
- Á quân tại cuộc thi ILSVRC 2014
- Sâu hơn rất nhiều so với AlexNet
- 140 triệu tham số

Input

3x3 conv, 64

3x3 conv, 64

Pool 1/2

3x3 conv, 128

3x3 conv, 128

Pool 1/2

3x3 conv, 256

3x3 conv, 256

Pool 1/2

3x3 conv, 512

3x3 conv, 512

3x3 conv, 512

Pool 1/2

3x3 conv, 512

3x3 conv, 512

3x3 conv, 512

Pool 1/2

FC 4096

FC 4096

FC 1000

Softmax

VGGNet

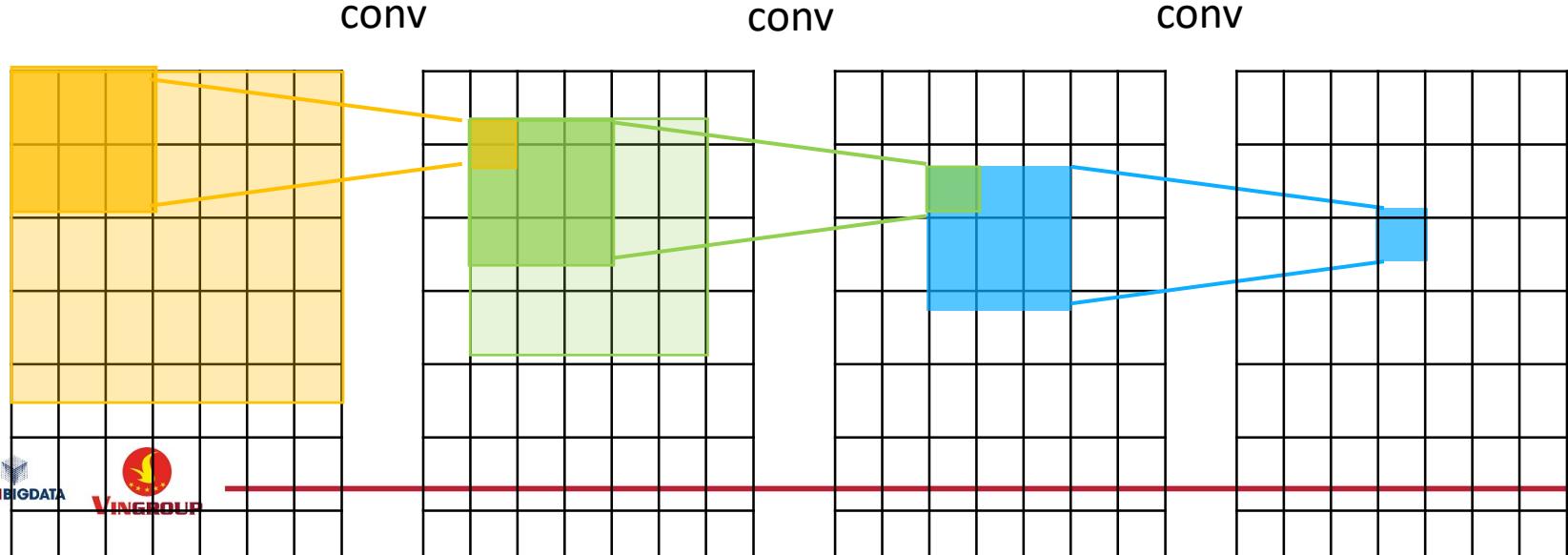
- Nơ-ron kích thước bé
Chỉ dùng conv 3x3, stride 1, pad 1 và 2x2 MAX POOL , stride 2
- Mạng sâu hơn
AlexNet: 8 lớp
VGGNet: 16 - 19 lớp
- ZFNet: 11.7% top 5 error in ILSVRC'13
- VGGNet: 7.3% top 5 error in ILSVRC'14

[Simonyan and Zisserman, 2014]

VGGNet

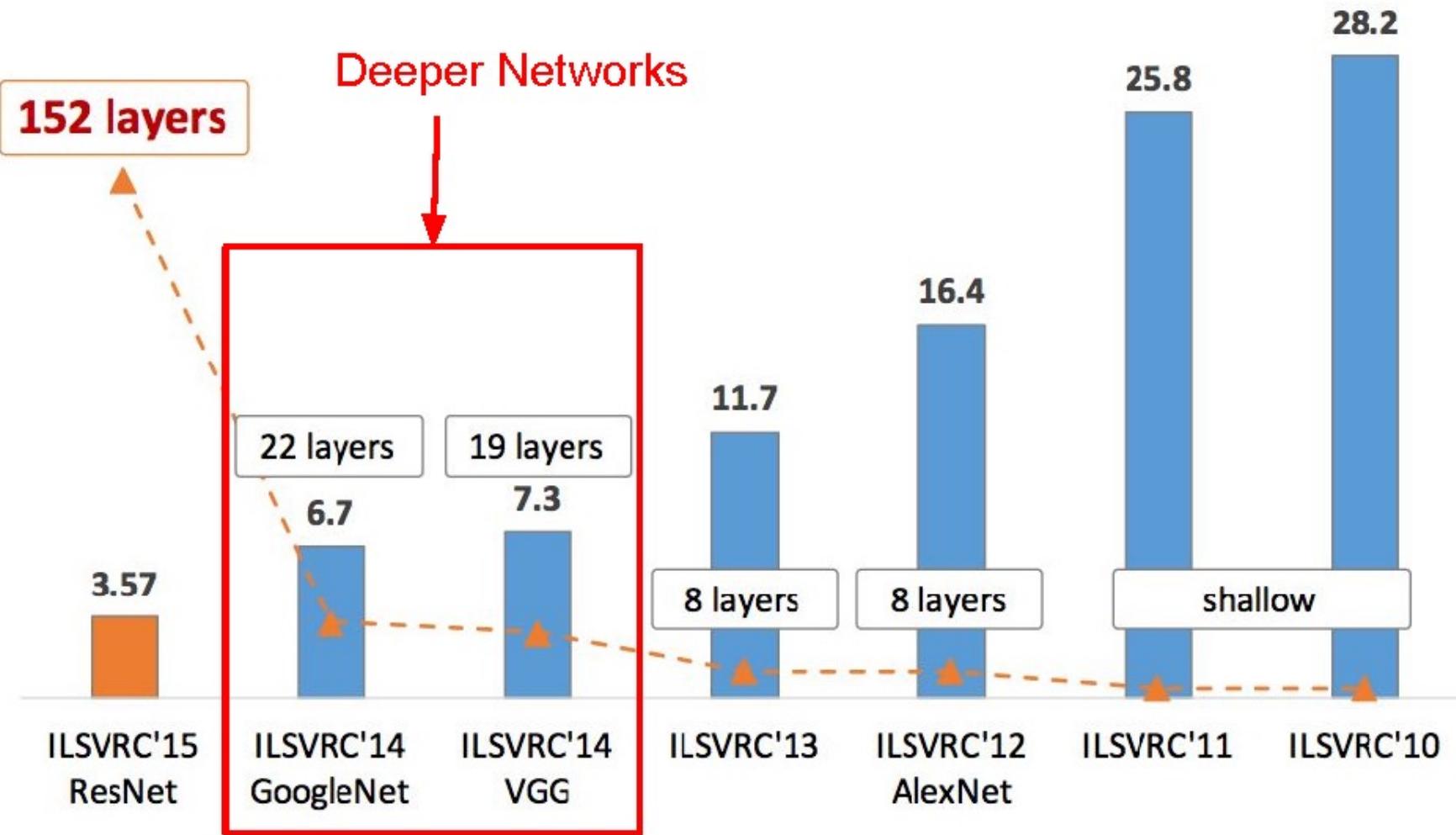
- Tại sao dùng filter bé? (3×3 conv)
- Chồng 3 lớp 3×3 conv (stride 1) có cùng hiệu quả thu nhận thông tin như một lớp 7×7 conv.
- Nhưng sâu hơn, nhiều lớp phi tuyến hơn
- Và ít tham số hơn: $3 * (3^2 C^2)$ vs. $7^2 C^2$ với C là số kênh của mỗi lớp

[Simonyan and Zisserman, 2014]



Input	memory: 224*224*3=150K	params: 0
3x3 conv, 64	memory: 224*224*64=3.2M	params: $(3*3*3)*64 = 1,728$
3x3 conv, 64	memory: 224*224*64=3.2M	params: $(3*3*64)*64 = 36,864$
Pool	memory: 112*112*64=800K	params: 0
3x3 conv, 128	memory: 112*112*128=1.6M	params: $(3*3*64)*128 = 73,728$
3x3 conv, 128	memory: 112*112*128=1.6M	params: $(3*3*128)*128 = 147,456$
Pool	memory: 56*56*128=400K	params: 0
3x3 conv, 256	memory: 56*56*256=800K	params: $(3*3*128)*256 = 294,912$
3x3 conv, 256	memory: 56*56*256=800K	params: $(3*3*256)*256 = 589,824$
3x3 conv, 256	memory: 56*56*256=800K	params: $(3*3*256)*256 = 589,824$
Pool	memory: 28*28*256=200K	params: 0
3x3 conv, 512	memory: 28*28*512=400K	params: $(3*3*256)*512 = 1,179,648$
3x3 conv, 512	memory: 28*28*512=400K	params: $(3*3*512)*512 = 2,359,296$
3x3 conv, 512	memory: 28*28*512=400K	params: $(3*3*512)*512 = 2,359,296$
Pool	memory: 14*14*512=100K	params: 0
3x3 conv, 512	memory: 14*14*512=100K	params: $(3*3*512)*512 = 2,359,296$
3x3 conv, 512	memory: 14*14*512=100K	params: $(3*3*512)*512 = 2,359,296$
3x3 conv, 512	memory: 14*14*512=100K	params: $(3*3*512)*512 = 2,359,296$
Pool	memory: 7*7*512=25K	params: 0
FC 4096	memory: 4096	params: $7*7*512*4096 = 102,760,448$
FC 4096	memory: 4096	params: $4096*4096 = 16,777,216$
FC 1000	memory: 1000	params: $4096*1000 = 4,096,000$

ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners



Slide taken from Fei-Fei & Justin Johnson & Serena Yeung. Lecture 9.

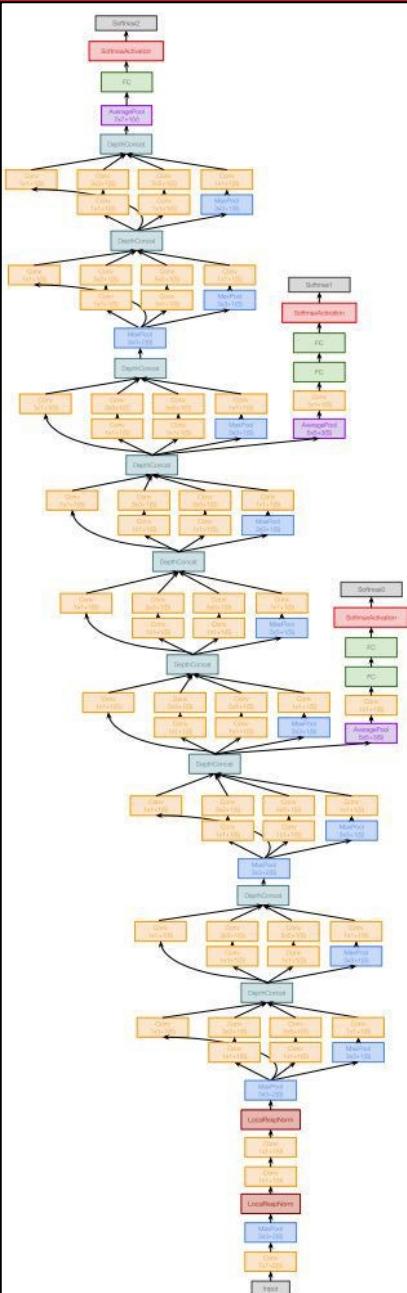
GoogleNet

- Going Deeper with Convolutions - Christian Szegedy et al.; 2015
- Vô địch ILSVRC 2014
- Sâu hơn nhiều so với AlexNet
- Số tham số ít hơn 12 lần so với AlexNet
- Tập trung vào giảm độ phức tạp tính toán

[Szegedy et al., 2014]

GoogleNet

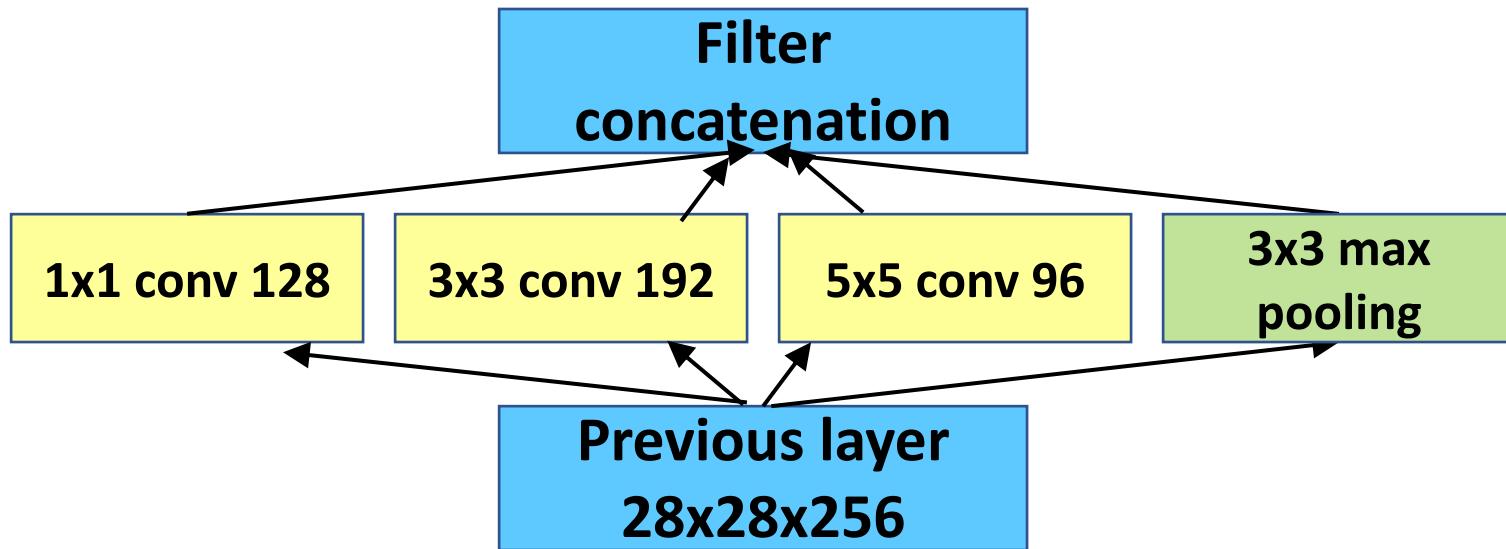
- 22 lớp
- Khối “Inception”
- Không có lớp kết nối đầy đủ (FC layers)
- Chỉ 5 triệu tham số!
- Vô địch tác vụ phân loại ảnh ILSVRC’14 (6.7% top 5 error)



[Szegedy et al., 2014]

GoogleNet - Naïve Inception Model

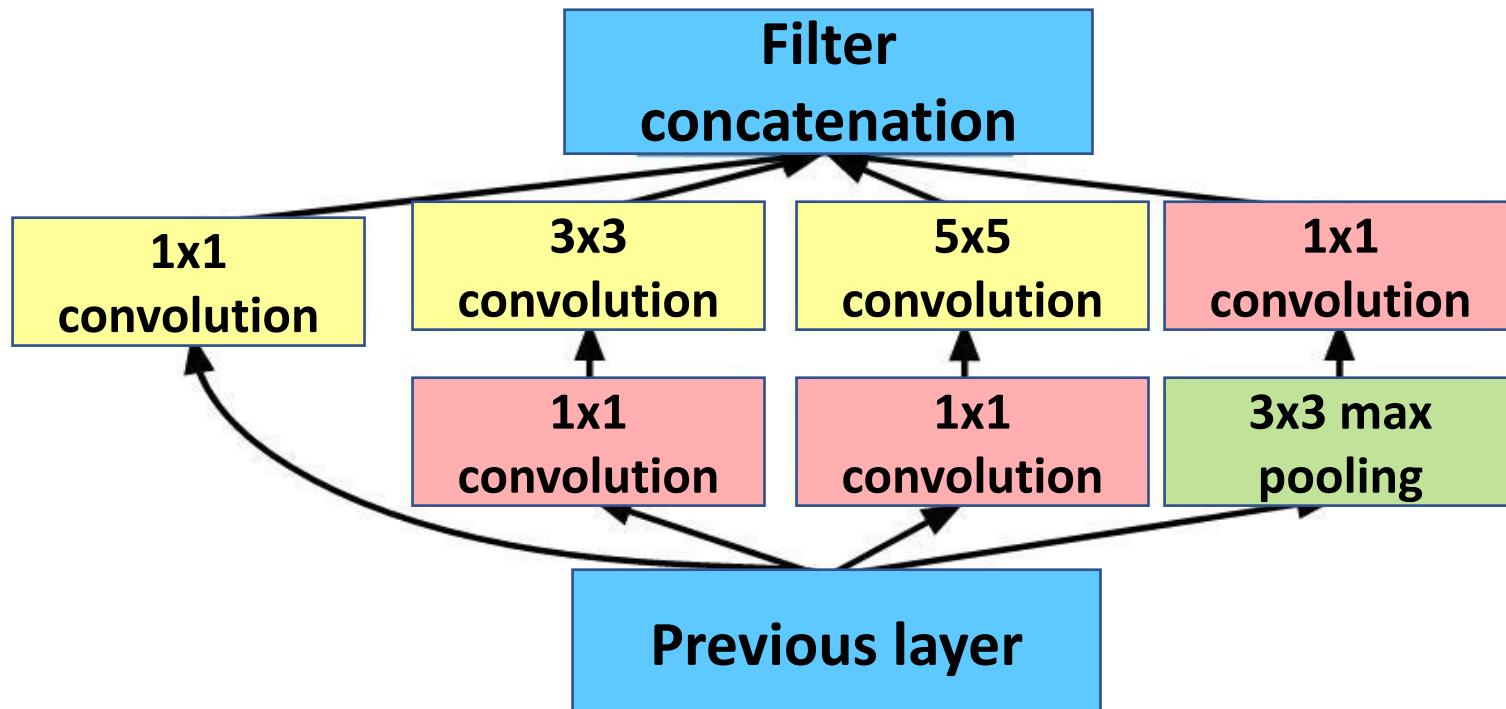
- Số lượng phép tích chập:
- 1x1 conv, 128: $28 \times 28 \times 128 \times 1 \times 1 \times 256$
- 3x3 conv, 192: $28 \times 28 \times 192 \times 3 \times 3 \times 256$
- 5x5 conv, 96: $28 \times 28 \times 96 \times 5 \times 5 \times 256$
- Tổng cộng: 854M ops ==> **Tính toán rất nặng!**



[Szegedy et al., 2014]

GoogleNet

- Giải pháp: lớp nút cỗ chai “bottleneck” sử dụng conv 1×1 để giảm chiều sâu khối dữ liệu.



[Szegedy et al., 2014]

- Số lượng phép toán tích chập:**

1x1 conv, 64: $28 \times 28 \times 64 \times 1 \times 1 \times 256$

1x1 conv, 64: $28 \times 28 \times 64 \times 1 \times 1 \times 256$

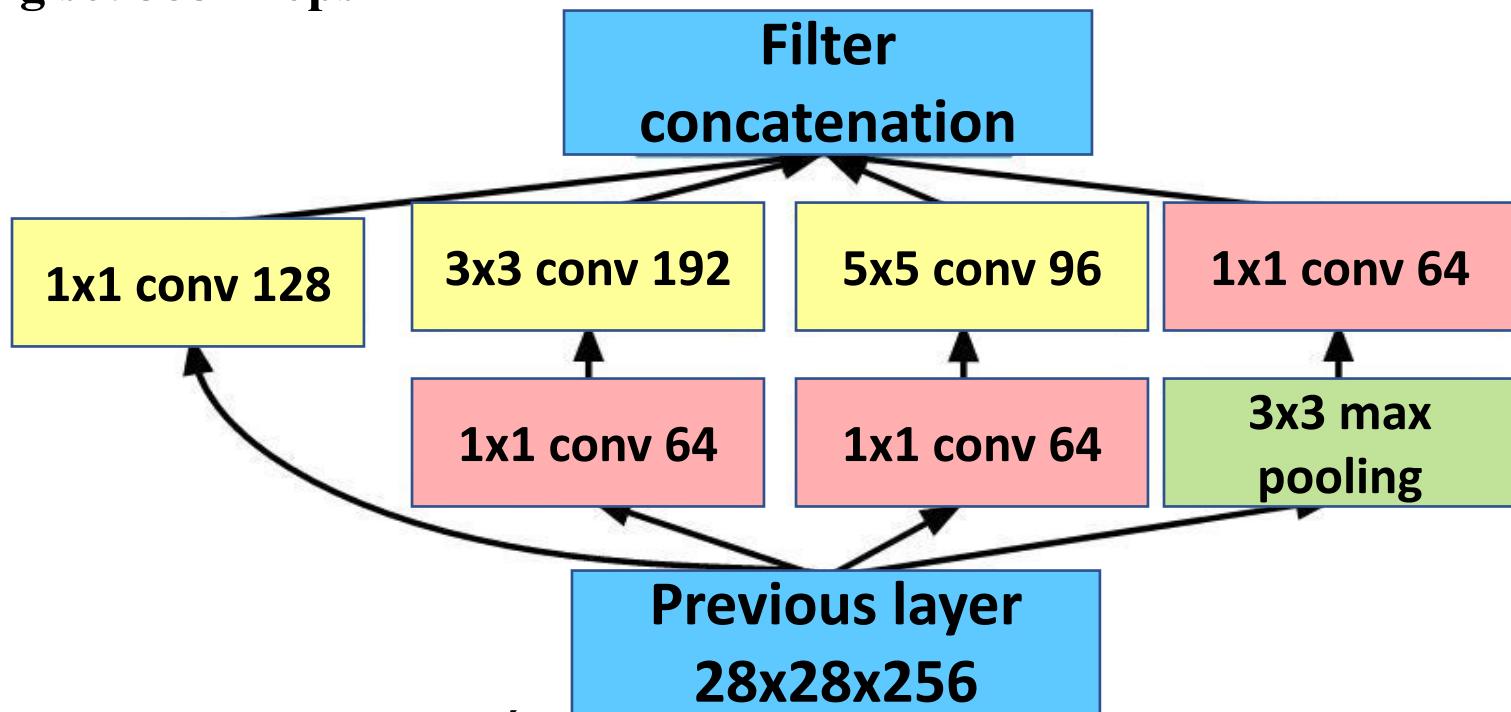
1x1 conv, 128: $28 \times 28 \times 128 \times 1 \times 1 \times 256$

3x3 conv, 192: $28 \times 28 \times 192 \times 3 \times 3 \times 64$

5x5 conv, 96: $28 \times 28 \times 96 \times 5 \times 5 \times 64$

1x1 conv, 64: $28 \times 28 \times 64 \times 1 \times 1 \times 256$

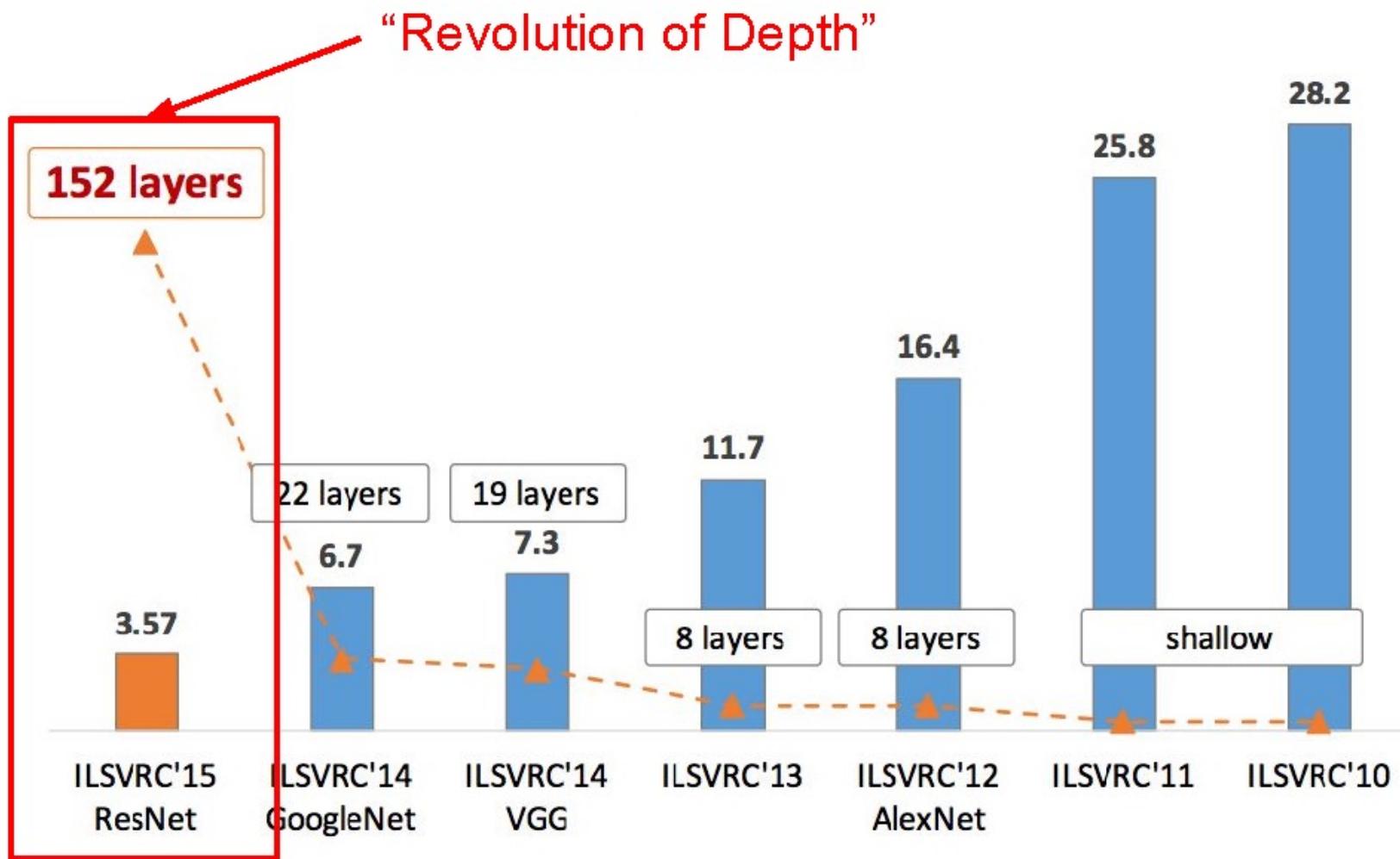
Tổng số: 353M ops



- So với 854M ops với khối inception thường

[Szegedy et al., 2014]

ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners



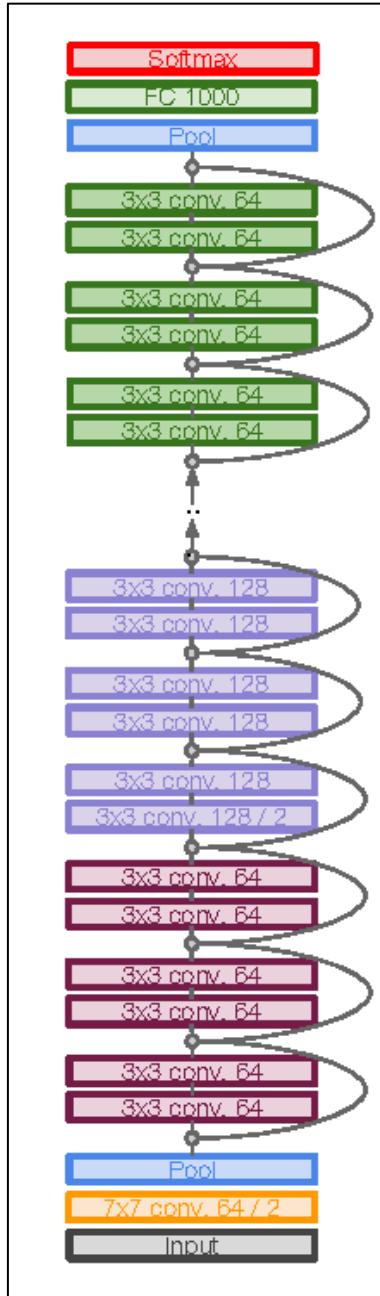
ResNet

- Deep Residual Learning for Image Recognition - Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun; 2015
- Mạng rất sâu, tới 152 lớp
- Mạng càng sâu càng khó huấn luyện.
- Mạng càng sâu càng chịu nhiều ảnh hưởng của vấn đề triệt tiêu và bùng nổ gradient.
- ResNet đề xuất phương pháp học phần dư (residual learning) cho phép huấn luyện hiệu quả các mạng sâu hơn rất nhiều so với các mạng xuất hiện trước đó.

[He et al., 2015]

ResNet

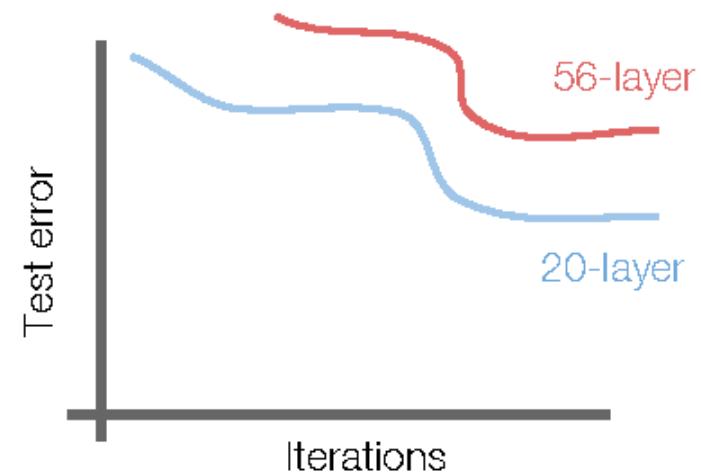
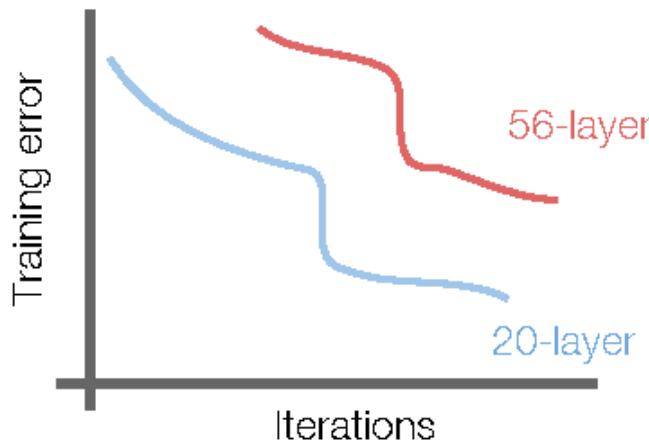
- Vô địch tác vụ phân loại ILSVRC'15 (3.57% top 5 error, trong khi sai số của con người khoảng 5.1%)
- Càn quét tất cả các cuộc thi về phân loại ảnh tại ILSVRC'15 và COCO'15!



[He et al., 2015]

ResNet

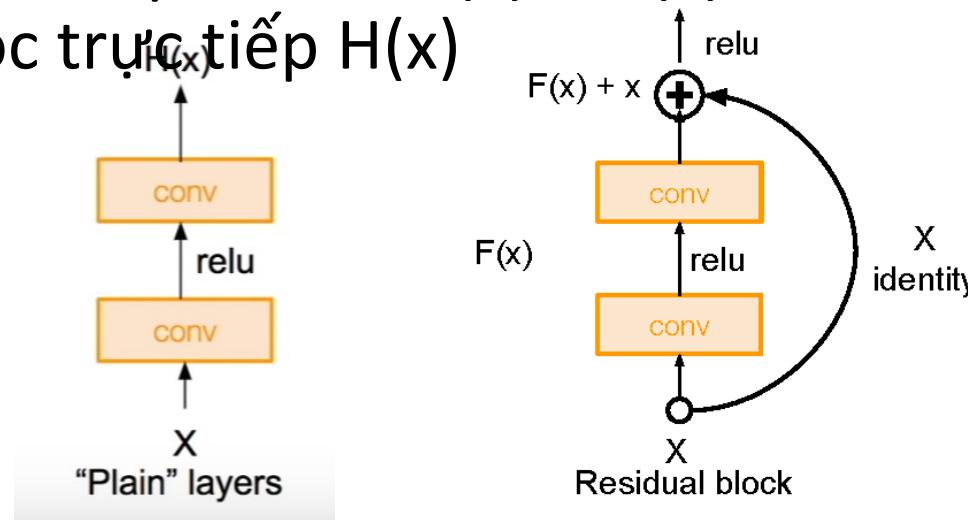
- Điều gì xảy ra khi chúng ta tăng độ sâu mạng nơ-ron?
- Mạng 56 lớp làm việc kém hơn cả trên tập huấn luyện lẫn tập test (không phải do overfitting gây ra)



[He et al., 2015]

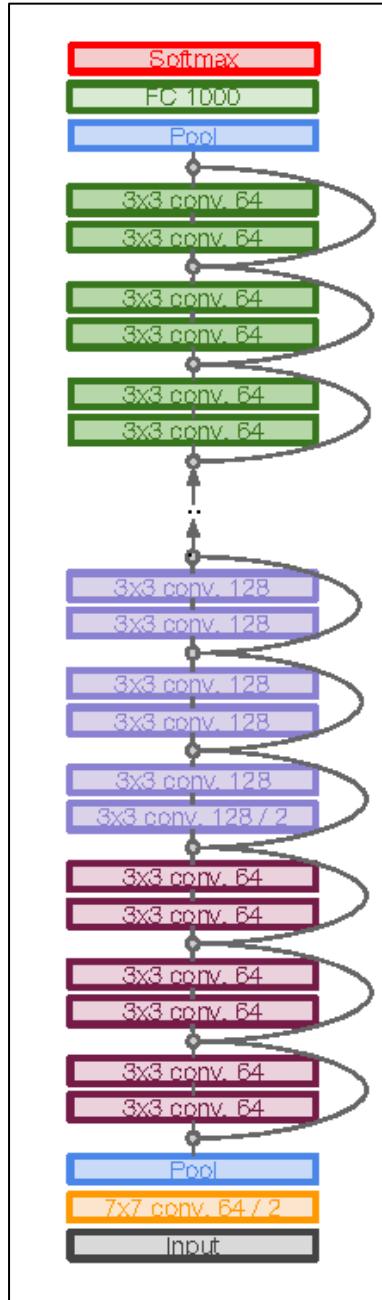
ResNet

- Giả thiết: Vấn đề ở chỗ bài toán tối ưu. Mạng rất sâu sẽ khó hơn để tối ưu.
- Giải pháp: Dùng các lớp mạng để học biểu diễn phần dư (sự sai khác giữa đầu ra và đầu vào) thay vì học trực tiếp đầu ra như trước.
- Học biểu diễn phần dư $F(x) = H(x) - x$ thay vì học trực tiếp $H(x)$



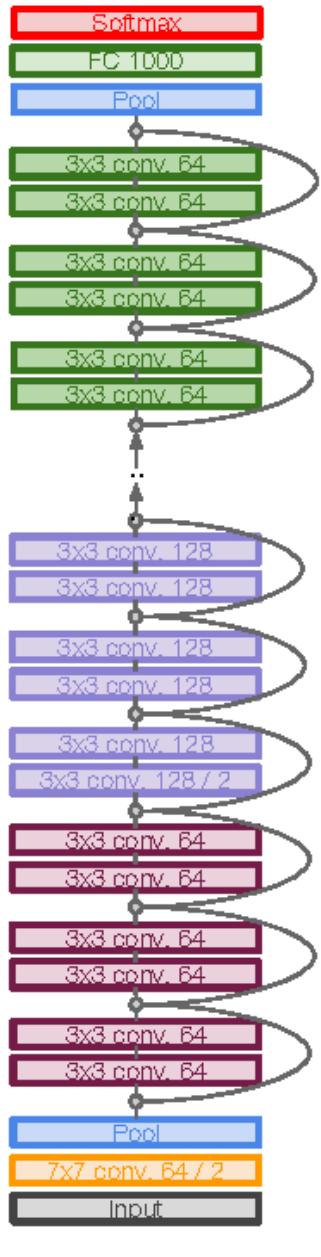
[He et al., 2015]

ResNet



- Kiến trúc ResNet đầy đủ:
- Chồng các khối phần dư residual blocks
- Mỗi khối có hai lớp 3x3 conv
- Định kỳ tăng gấp đôi số lượng filter và giảm độ phân giải bằng conv bước nhảy stride 2
- Lớp conv phụ ở đầu mạng
- Không có lớp FC ở cuối (chỉ có lớp FC 1000 để xuất ra kết quả phân loại 1000 lớp)

[He et al., 2015]

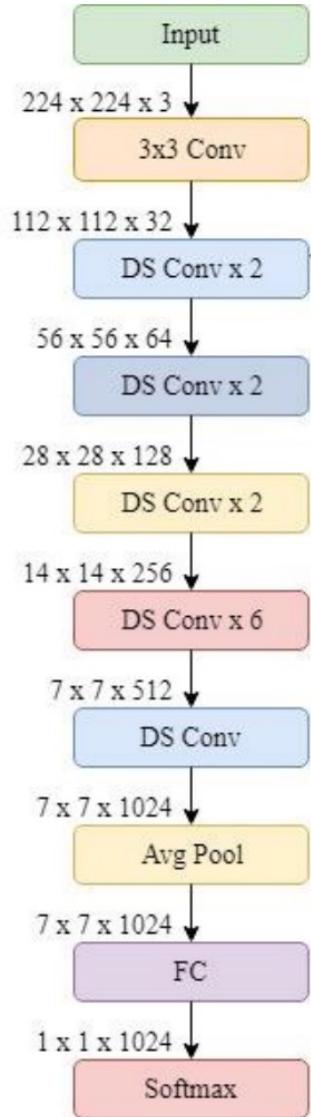


ResNet

- Độ sâu của mạng khi tham gia cuộc thi ImageNet: 34, 50, 101, 152
- Với các mạng sâu (ResNet-50+), tác giả dùng lớp “bottleneck” để tăng hiệu quả (tương tự như GoogLeNet)

[He et al., 2015]

MobileNets V1



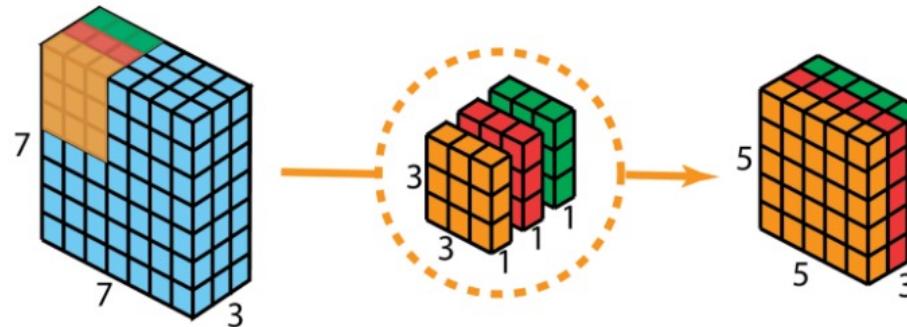
- Thiết kế mạng CNN tối ưu cho môi trường thiết bị di động, thiết bị nhúng
 - Năng lực tính toán hạn chế
 - Thời gian suy diễn cần nhanh
- Sử dụng Deepwise separable convolutions
- Đề xuất 2 siêu tham số toàn cục để cân nhắc đánh đổi giữa độ trễ và độ chính xác

Table 8. MobileNet Comparison to Popular Models

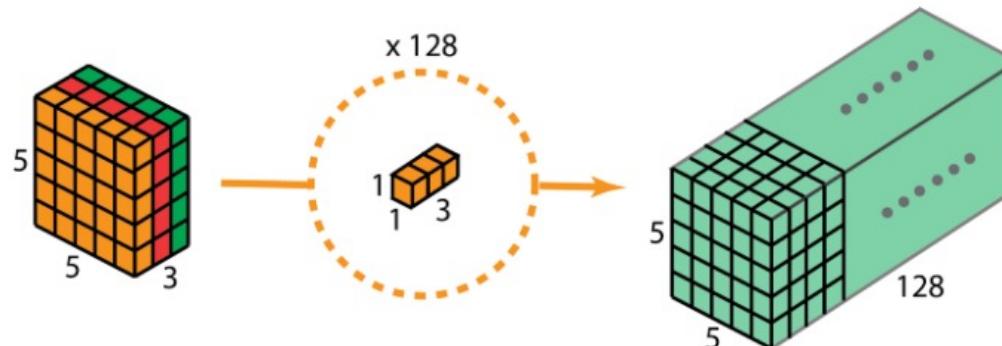
Model	ImageNet	Million	Million
	Accuracy	Mult-Adds	Parameters
1.0 MobileNet-224	70.6%	569	4.2
GoogleNet	69.8%	1550	6.8
VGG 16	71.5%	15300	138

Deepwise separable convolution

- Phân rã 1 phép tích chập thành 1 depthwise convolution và 1 1x1 convolution (pointwise convolution)
- Depthwise convolution $3 \times 3 \times 3 \times 1 \times 5 \times 5 = 675$ multiplications

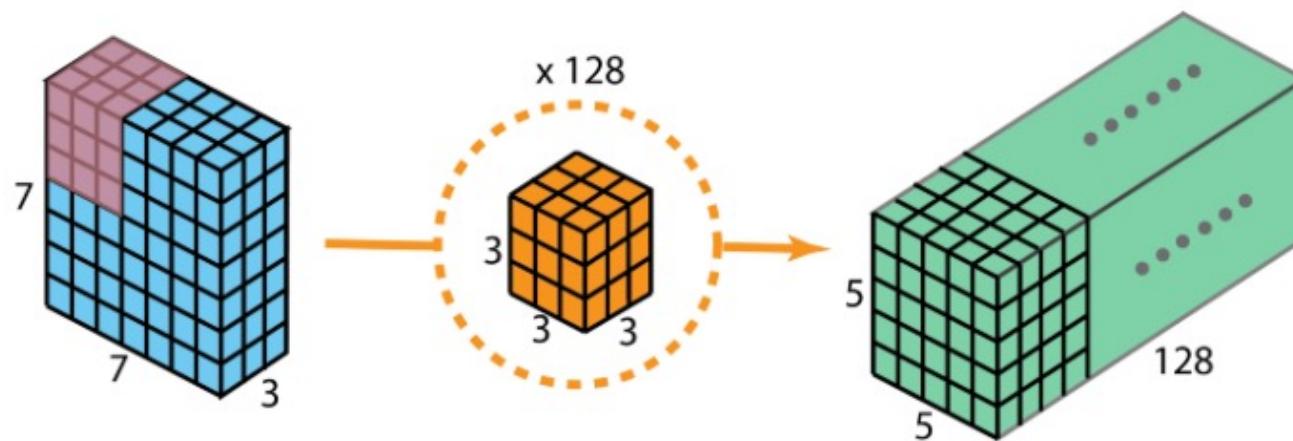


- Pointwise convolution $128 \times 1 \times 1 \times 3 \times 5 \times 5 = 9,600$ multiplications



So sánh với phép tích chập thông thường

- 128 3x3x3 filters trượt 5x5 lần
- Số lượng các phép nhân nhiều hơn
- Số lượng các tham số cần học nhiều hơn

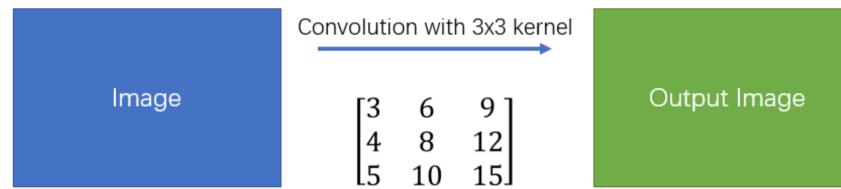


$$128 \times 3 \times 3 \times 3 \times 5 \times 5 = 86,400 \text{ multiplications}$$

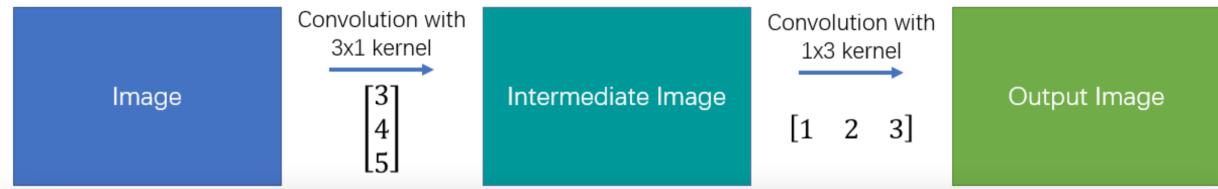
Spatially separable convolution

- Thường được gọi là “separable convolution”, phổ biến trong cộng đồng xử lý ảnh (image processing)
- Không được sử dụng phổ biến trong các mạng học sâu
- Nó phân tách phép toán tích chập thành hai phần và áp dụng từng tích chập riêng biệt liên tiếp.
- Cho phép giảm số lượng phép nhân, giảm độ phức tạp tính toán

Simple Convolution



Spatial Separable Convolution

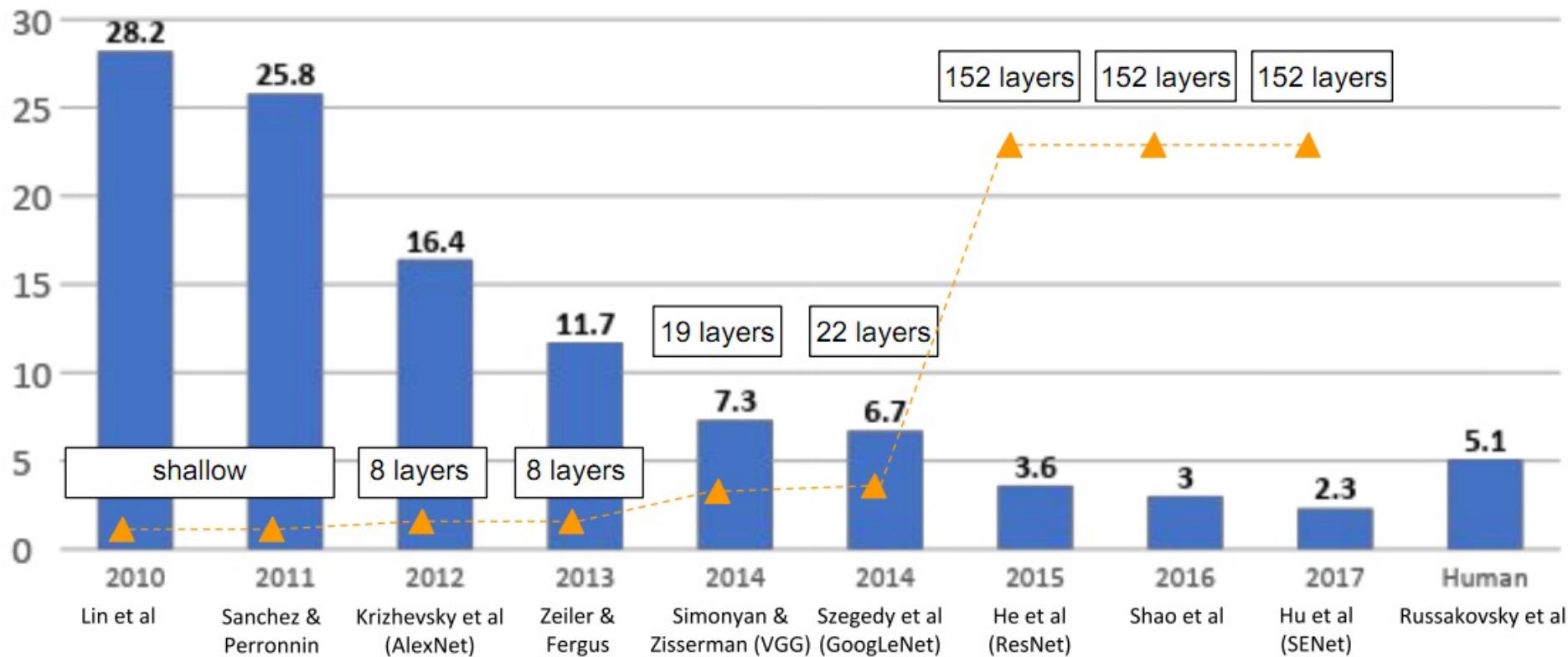


Width Multiplier và Resolution Multiplier

- Với kiến trúc MobileNet cơ sở đã đề xuất, có thể tạo ra các mạng có kiến trúc nhỏ hơn
- Width Multiplier $\alpha \in (0, 1]$, kích thước kênh đầu vào và đầu ra được xác định bằng αM và αN
 - Có hiệu ứng giảm chi phí tính toán khoảng α^2
 - Chi phí tính toán với Conv dw
 - $D_K \cdot D_K \cdot \alpha M \cdot D_F \cdot D_F + \alpha M \cdot \alpha N \cdot D_F \cdot D_F$
- Resolution multiplier $\rho \in (0, 1]$, giảm kích thước ảnh đầu vào và trung gian bởi ρ
 - Có hiệu ứng giảm chi phí tính toán ρ^2
- Chi phí tính toán sau khi áp dụng α và ρ
 - $D_K \cdot D_K \cdot \alpha M \cdot \rho D_F \cdot \rho D_F + \alpha M \cdot \alpha N \cdot \rho D_F \cdot \rho D_F$

Recent SOTA

ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners



Slide taken from Fei-Fei & Justin Johnson & Serena Yeung. Lecture 9.

Squeeze-and-Excitation Networks (SENet)

- Phép tích chập thông thường xây dựng các đặc trưng bằng việc kết hợp các thông tin ở cả chiều không gian (spatial) và kênh (channel).
- SENet đề xuất các khối “Squeeze-and-Excitation” (SE) cho phép thu nhận, mô hình hoá, học được các đặc trưng tương minh về mối quan hệ tương quan giữa các kênh.

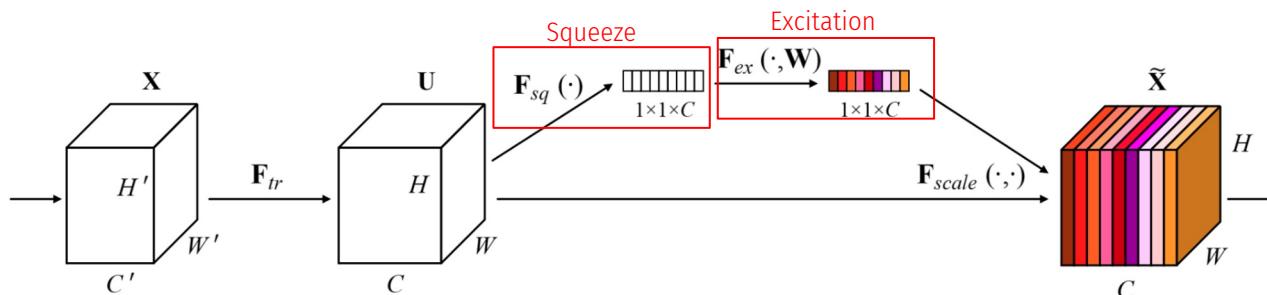


Fig. 1. A Squeeze-and-Excitation block.

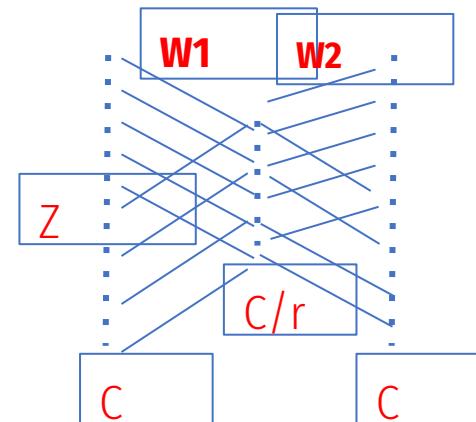
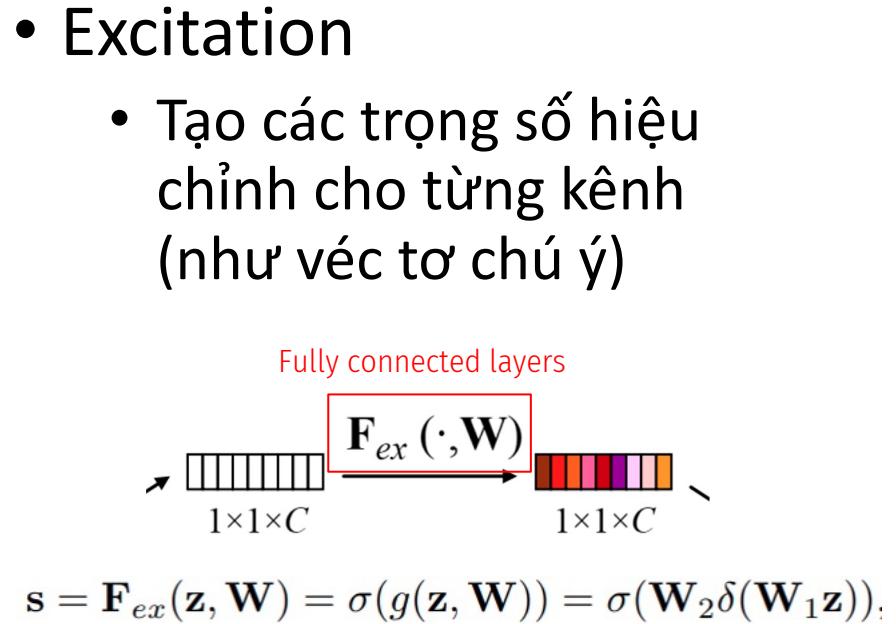
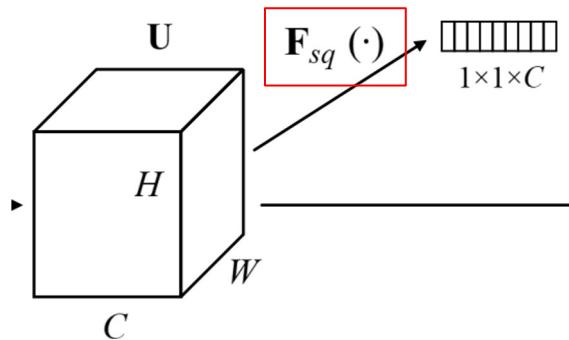
$$\tilde{\mathbf{x}}_c = \mathbf{F}_{scale}(\mathbf{u}_c, s_c) = s_c \mathbf{u}_c$$

Squeeze and Excitation block

- Squeeze
 - Tạo bộ mô tả kênh bằng cách tổng hợp các bản đồ đặc trưng (feature maps) theo các chiều không gian ($H \times W$)
- Excitation
 - Tạo các trọng số hiệu chỉnh cho từng kênh (như véc tơ chú ý)

$$z_c = \mathbf{F}_{sq}(\mathbf{u}_c) = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W u_c(i, j).$$

Global average pool



Squeeze and Excitation block

- Có thể dễ dàng áp dụng vào các kiến trúc mạng phổ biến

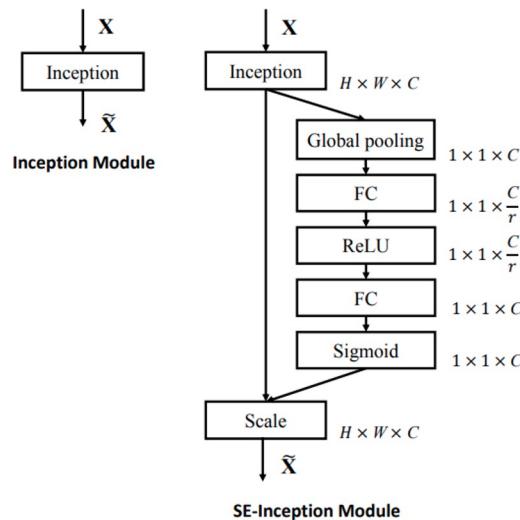
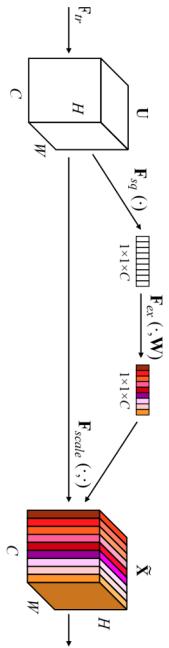


Fig. 2. The schema of the original Inception module (left) and the SE-Inception module (right).

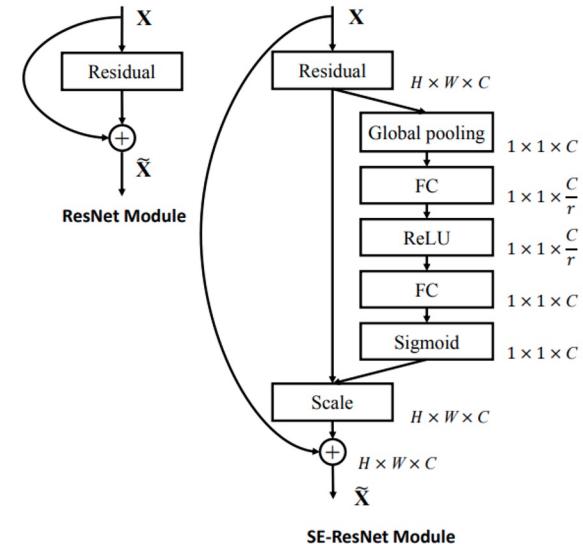


Fig. 3. The schema of the original Residual module (left) and the SE-ResNet module (right).

SENet: Đánh giá hiệu năng

TABLE 2

Single-crop error rates (%) on the ImageNet validation set and complexity comparisons. The *original* column refers to the results reported in the original papers (the results of ResNets are obtained from the website: <https://github.com/Kaiminghe/deep-residual-networks>). To enable a fair comparison, we re-train the baseline models and report the scores in the *re-implementation* column. The *SENet* column refers to the corresponding architectures in which SE blocks have been added. The numbers in brackets denote the performance improvement over the re-implemented baselines. † indicates that the model has been evaluated on the non-blacklisted subset of the validation set (this is discussed in more detail in [21]), which may slightly improve results. VGG-16 and SE-VGG-16 are trained with batch normalization.

	original		re-implementation			SENet		
	top-1 err.	top-5 err.	top-1 err.	top-5 err.	GFLOPs	top-1 err.	top-5 err.	GFLOPs
ResNet-50 [13]	24.7	7.8	24.80	7.48	3.86	23.29 _(1.51)	6.62 _(0.86)	3.87
ResNet-101 [13]	23.6	7.1	23.17	6.52	7.58	22.38 _(0.79)	6.07 _(0.45)	7.60
ResNet-152 [13]	23.0	6.7	22.42	6.34	11.30	21.57 _(0.85)	5.73 _(0.61)	11.32
ResNeXt-50 [19]	22.2	-	22.11	5.90	4.24	21.10 _(1.01)	5.49 _(0.41)	4.25
ResNeXt-101 [19]	21.2	5.6	21.18	5.57	7.99	20.70 _(0.48)	5.01 _(0.56)	8.00
VGG-16 [11]	-	-	27.02	8.81	15.47	25.22 _(1.80)	7.70 _(1.11)	15.48
BN-Inception [6]	25.2	7.82	25.38	7.89	2.03	24.23 _(1.15)	7.14 _(0.75)	2.04
Inception-ResNet-v2 [21]	19.9†	4.9†	20.37	5.21	11.75	19.80 _(0.57)	4.79 _(0.42)	11.76

EfficientNet

- Sử dụng phương pháp tìm kiếm kiến trúc mạng nơ ron tối ưu (neural architecture search) để có mạng cơ sở EfficientNet-B0
- Đề xuất kỹ thuật mở rộng, thu nhỏ (scaling) kích thước mạng sử dụng một hệ số đơn giản, hợp nhất cho cả 3 chiều: chiều sâu, chiều rộng và độ phân giải.

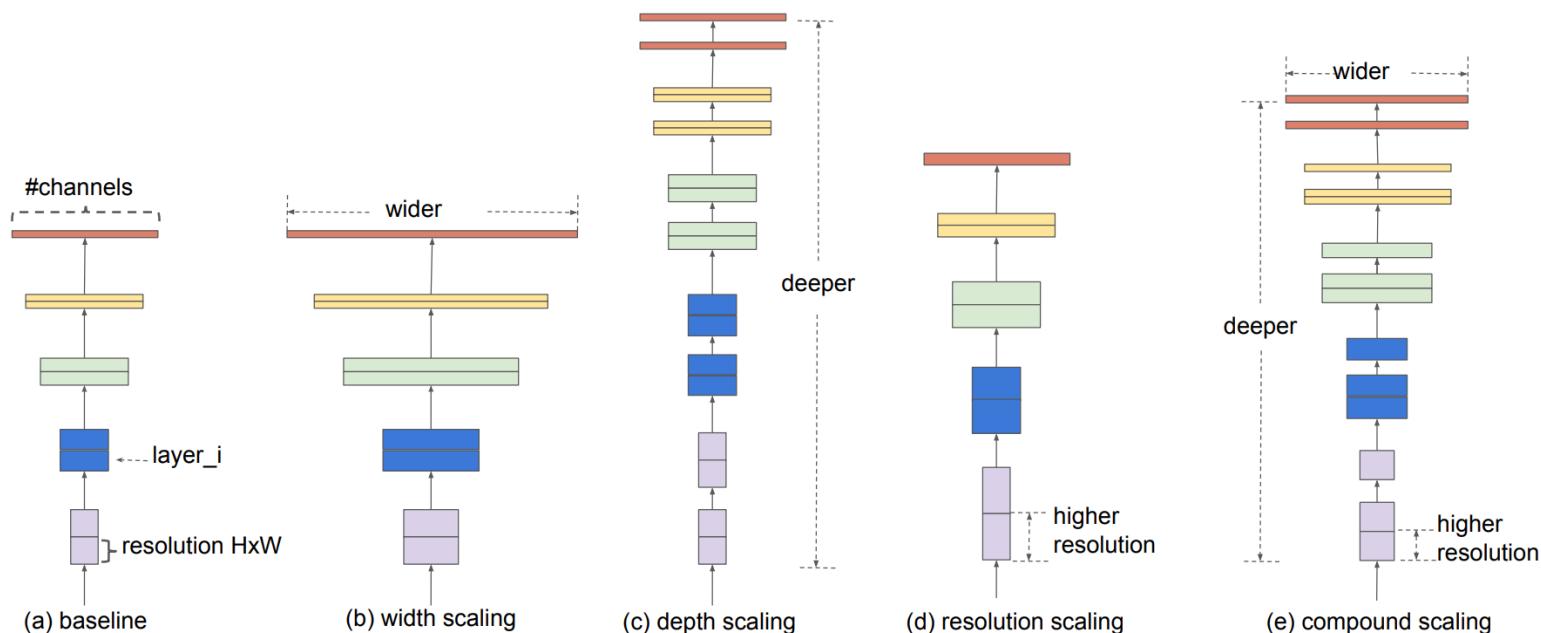
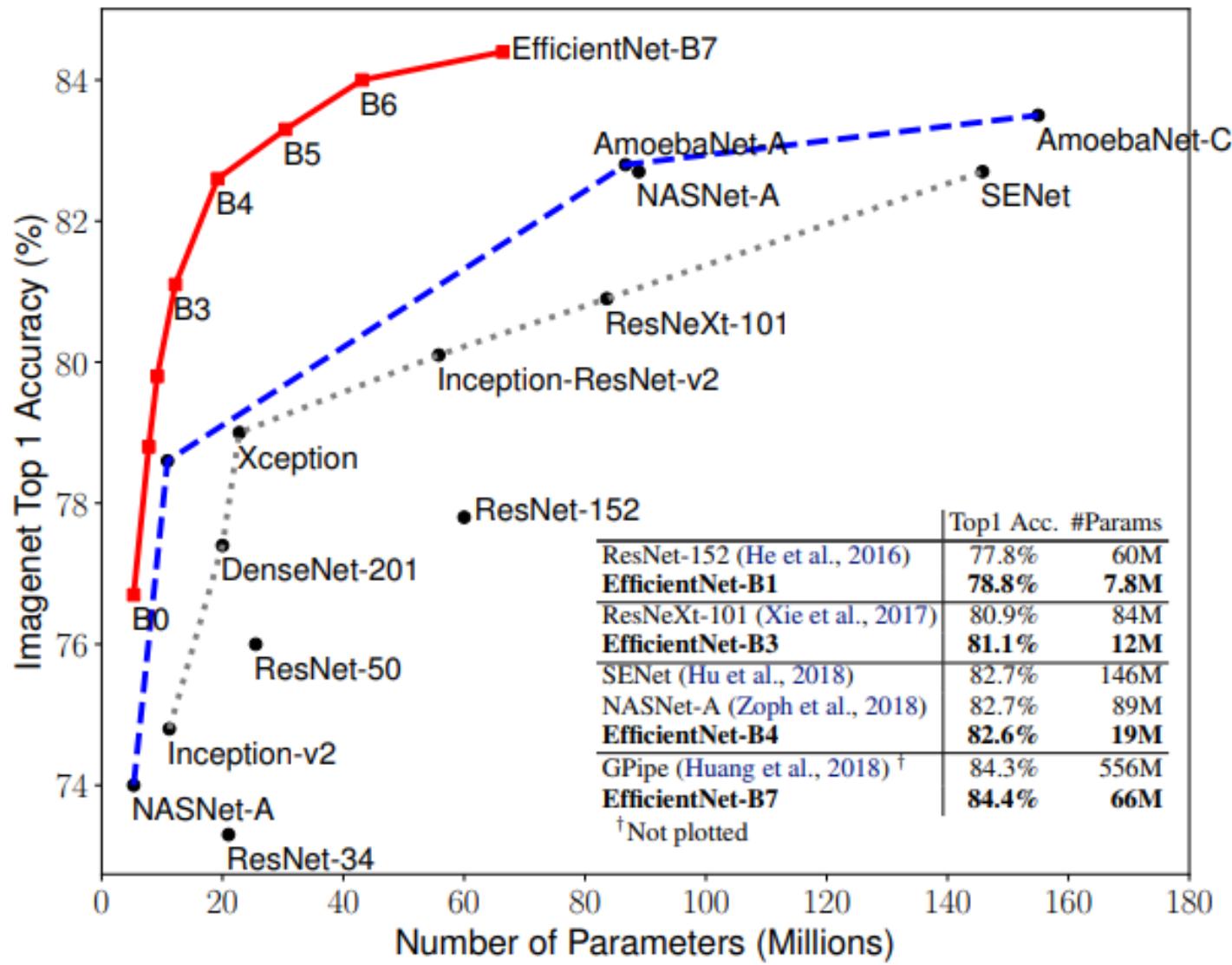
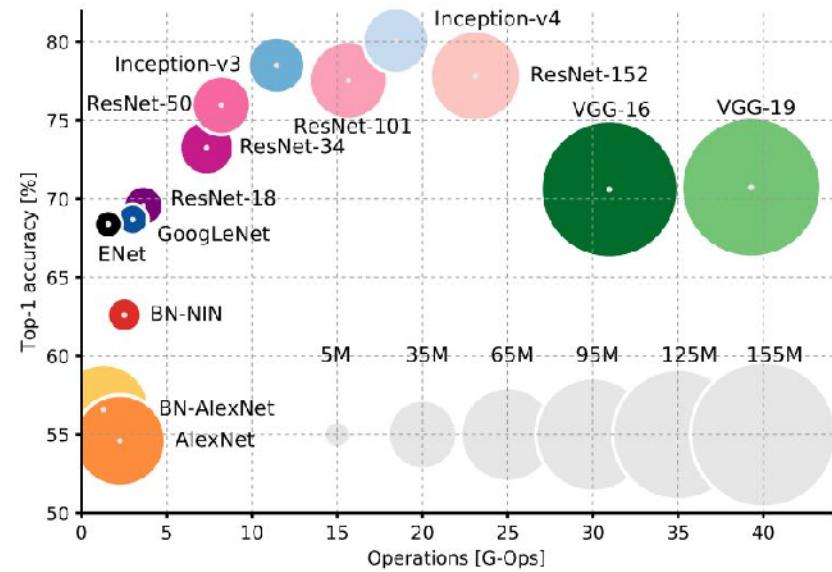
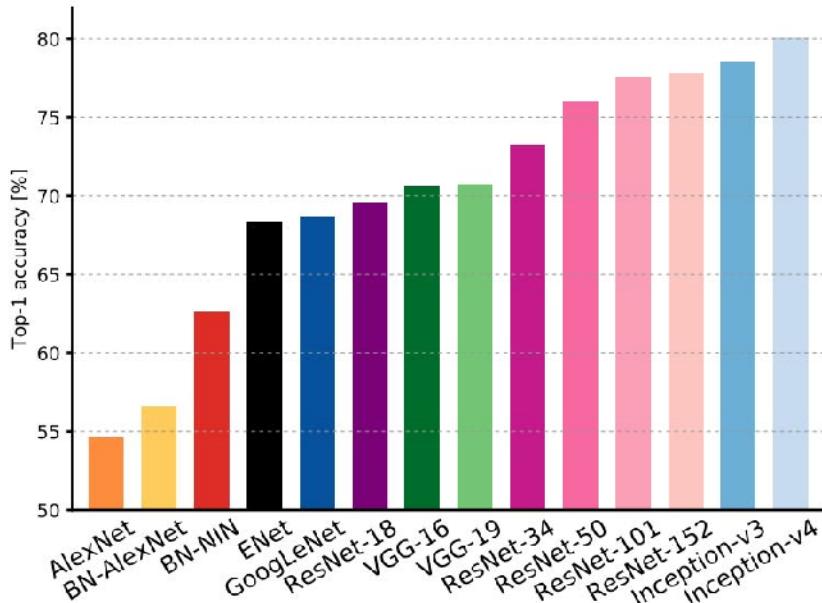


Figure 2. Model Scaling. (a) is a baseline network example; (b)-(d) are conventional scaling that only increases one dimension of network width, depth, or resolution. (e) is our proposed compound scaling method that uniformly scales all three dimensions with a fixed ratio.

Recent SOTA



Accuracy comparison



Tài liệu tham khảo

1. Khóa học Intro to DL của MIT:

<http://introtodeeplearning.com/>

2. Khóa học cs231n của Stanford:

<http://cs231n.stanford.edu/>